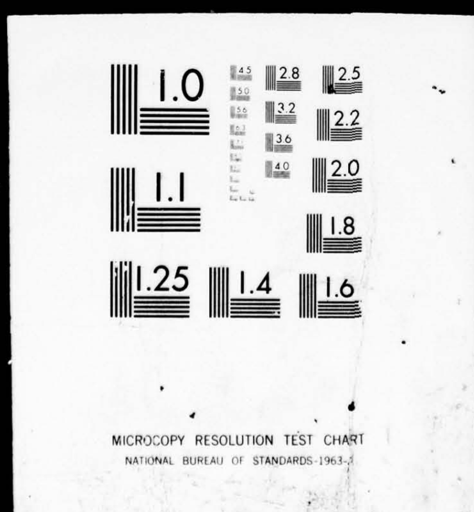
AD-A068 266 PURDUE UNIV LAFAYETTE IND PURDUE LAB FOR APPLIED IND--ETC F/G 9/2
MINUTES 1978 SPRING REGIONAL MEETING INTERNATIONAL PURDUE WORKS--ETC(U)
JUN 78 N00014-78-C-0127

UNCLASSIFIED NL

1 OF 4
AD
A068266

1 OF 4

AD
A068266



MINUTES

1978 SPRING REGIONAL [illegible]
INTERNATIONAL PURDUE WO[illegible]
ON INDUSTRIAL COMPUTER [illegible]

PART I

NARRATIVE AND TECHNICAL [illegible]

FEDERAL INSTITUTE OF TECHNOLOGY [illegible]

ZURICH, SWITZERLAND

April 4 – 7, 1978

PURDUE UNIVERSITY, WEST LAFAYETTE, [illegible]

April 10 – 12, 1978

JEIDA, TOKYO, JAPAN

June 28 – 30, 1978

Purdue Laboratory for Applied Industrial Control
Schools of Engineering
Purdue University
West Lafayette, Indiana 47907

This Report is Published as Part of
Engineering Experiment Station Bulletin 143 Series

Schools of Engineering
Purdue University
West Lafayette, Indiana 47907

(12) 3Ø7p.

(6) MINUTES (12)

1978 SPRING REGIONAL MEETING

INTERNATIONAL PURDUE WORKSHOPS

ON INDUSTRIAL COMPUTER SYSTEMS.

PART I.

NARRATIVE AND TECHNICAL APPENDICES.

Contract N00014-78-C-0127

held at FEDERAL INSTITUTE OF TECHNOLOGY (ETH)

ZURICH, SWITZERLAND

April 4 - 7, 1978,

PURDUE UNIVERSITY, WEST LAFAYETTE, INDIANA

April 1Ø - 12, 1978

and

JEIDA, TOKYO, JAPAN

June 29 - 3Ø, 1978.

(11) 3Ø Jun 78

(15) NØØØ14-78-C-Ø127

Purdue Laboratory for Applied Industrial Control
Schools of Engineering
Purdue University
West Lafayette, Indiana 47907

408 244

JM

The International Purdue Workshop on
Industrial Computer Systems
is
Jointly sponsored by the
Purdue Laboratory for Applied Industrial Control;
by the Automatic Control Systems Division,
the Chemical and Petroleum Industries Division,
and the Data Handling and Computations Division
of the
Instrument Society of America;

by the
Japan Electronic Industry Development Association
(JEIDA) through the IPW Japan Committee;
and by the Commission of the European Communities
through its General Directorate for Internal Markets
and Industrial Affairs.

It is also sponsored by the
International Federation for Information Processing
as Working Group 5.4 of Technical Committee TC-5,
and by the
Associate Committee for Automatic Control,
National Research Council of Canada.

The work of the International Purdue Workshop
is
partially funded by a grant from the
Naval Air Systems Command, U.S. Navy,
through the Office of Naval Research,
Washington, D.C.

The Workshop is affiliated with the
Institute of Electrical and Electronic Engineers
through the
Data Acquisition and Control Committee
of the Computer Society
and the
Industrial Control Committee
of the Industrial Applications Society.

It is also affiliated with the
International Federation of Automatic Control
through its
Computers Committee.

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

SECTION I

MINUTES OF THE EUROPEAN REGIONAL MEETING INTERNATIONAL PURDUE WORKSHOP ON INDUSTRIAL COMPUTER SYSTEMS

MINUTES OF THE EUROPEAN REGIONAL MEETING

INTERNATIONAL PURDUE WORKSHOP ON INDUSTRIAL COMPUTER SYSTEMS

Federal Institute of Technology (ETH)
Zurich, Switzerland
April 4 - 7, 1978

1. GENERAL

The annual regional meeting was held at ETH Zurich again, perfectly organized by Thierry Lalive d'Epinay. The environment has changed a little: there are new lecture rooms fitted in to the old university buildings, and - most important - a roof-top restaurant which was the place for a splendid social evening. The Agenda of Insert E-1 was used. Those listed in Appendix E-1 were registered.

Approximately 95 attendees from 16 countries were present. The meeting lasted from April 4 to April 7, 1978. On April 3 the "steering committee" (i.e., Nicolas Malagardis and the TC chairmen) met together with Ken Thompson from the CEC.

The true highlights of this 5th Regional Meeting were a set of high-level tutorials, namely:

a. W. Whitaker's presentation of the US-DoD High Order Language Project.

b. N. Malagardis' tutorial on standardization.

c. L. Pouzin, who made a speech on Networks (Who else would be as competent as he?); and

d. D. Bellanger gave a report both on design and on real experience in robotics.

As usual the state of Purdue Europe was shown in presentations of PE chairman and the TC chairmen; in parallel sessions both independent and various joint TC meetings were held.

In the closing session various interesting topics were discussed in the plenum:

FORTRAN: As the proposal of Purdue Europe TC 1 concerning FORTRAN - extensions towards parallel programming and data management is different from the American S61.3 draft, the assembly asked TC 1 to try to achieve an agreement before a proposal is handed over to any official standardization body.

SIRE: The extended serial interface proposal seems to be a well designed protocol and bus system for industrial applications. Previous problems like, e.g., "master switching", have been resolved.

INSERT E-I

AGENDA

FIFTH EUROPEAN REGIONAL MEETING
INTERNATIONAL PURDUE WORKSHOP
ON INDUSTRIAL COMPUTER SYSTEMS

Federal Institute of Technology (ETH)

Zurich, Switzerland

April 4 - 7, 1978

<u>Tuesday, April 4th</u>

8.30 Registration

9.15 Introduction by the P.E. Chairman

Overview of Work Done in the Past Year

10.15 Coffee

10.45 Presentation of Results by TC Chairmen

12.30 Lunch

14.00 Parallel and/or Joint TC meetings

20.00 Presentation of DOD proposals

close

<u>Wednesday, April 5th</u>

9.00 Tutorial on Standards Methods and Practices
N. E. MALAGARDIS

10.00 Coffee

10.30 Tutorial on Private Networks and Industrial Uses
L. POUZIN

12.30 Lunch

14.00 Parallel and/or Joint TC meetings

close

18.00 Social Evening

Thursday, April 6th

9.00 Tutorial Experience with Real Time Systems

10.00 Coffee

10.30 Round Table on Real Time Operating Systems

12.30 Lunch

14.00 Discussion on TC 1 Real Time FORTRAN Proposal to ISO TC 97/SC 5/WG 1

Discussion on TC 5 SIRE Proposal

16.30 Closing Session

Election of Chairmen

Announcements

Motions

18.00 Close

Friday, April 7th

9.00 TC Parallel Sessions

12.00 Lunch

14.00 TC Parallel Sessions

17.00 Close

Operating Systems: TC 8's work was presented (in written form and as a lecture) and discussed. The assembly passed a motion concerning cooperation between TC 8 and other TCs:

2. <u>REPORTS OF THE TECHNICAL COMMITTEES</u>

TC 1 (Industrial Real Time FORTRAN):

At the Purdue Europe Regional Meeting the main paper "Industrial Real-Time FORTRAN" of Purdue Europe TC 1 was presented for discussion. The paper gained considerable interest and many valuable comments were given. The European Workshop also took into consideration the existence of the American standards and proposals on industrial real-time FORTRAN, i.e., S61.1, S61.2, and S61.3, and strongly emphasized that the differences between the American and European papers should be eliminated.

There are two ways to reach that goal:

(1) Complete correspondence

(2) For certain sections the European proposals may contain a correspondent subset of the American proposals and vice versa for other sections.

For the section on file handling, solution (2) could already be reached at the Regional Meeting with the future American paper S61.2 being a subset of the European proposal. For the section on multiprogramming

and real-time features, a similar agreement may be achieved by the participation of Mr. Caro of the American TC 1 in the next meetings of Purdue Europe TC 1.

TC 2 (Industrial Real Time BASIC):

The final proposal of a "Real-Time Enhancement to Minimal BASIC" which will be handed over to ANSI and ECMA in autumn '78 was discussed, amended, and is in good state. The cooperation with the standardization bodies has been further enlarged: We got the "order" to define the "Error Control Enhancement to Minimal BASIC" as well. This work will be influenced also by facts outside BASIC: TC 7, TC 8 and EDISG will have an impact.

TC 3 (Long Term Procedural Language):

The LTPL-E sub-group presented their evaluation of the USDoD high order language proposals, emphasizing parallelism, exception handling, I/O, algorithmic and descriptive facilities. The USDoD recorded their thanks for this and previous LTPL-E input, and wished to see continuing cooperation with the next phases of language development and the establishment of a suitable language environment.

Given that the LTPL-E project will not now proceed, and recognizing the importance of the USDoD HOL work, the committee reviewed its activities so as to allow

it to respond to the next phase of the HOL project. Essentially work is now being concentrated into 4 immediate areas -

- A report to the Commission of European Communities, reflecting the consensus view of the European (industrially oriented) evaluation team.
- Requirements for the environment of the language.
- Completion of existing I/O work.
- Test cases and examples to aid evaluation. (Longer term objectives will be given secondary priority at present).

These four areas will be enhanced further at the next meeting in Brussels from 31/5/78 - 2/6/78.

The committee recorded a unanimous vote of thanks to Peter Elzer who has resigned his chairmanship in order to take part in the USDoD HOL Team. The new chairman is D. N. Shorter, British Steel Corporation, 140 Battersea Park Road, London SW 11, England.

TC 4 (Problem Oriented Languages):

TC 4 continued the analysis and documentation of requirements for Problem Oriented Languages (POL) in two ways:

- User requirements on POLs as special application systems.
- Requirements on POLs as high order languages, which enable the user to program special

application systems or modules that are reusable in classes of applications.

With the aim to study the suitability of existing programming languages TC 4 had a presentation on the experiences with Real-Time FORTRAN using it to develop a software package for DNC. This presentation also gave a comparison of Real-Time FORTRAN and PEARL.

TC 5 (Interface and Data Transmission):

Highlights: Presentation of the SIRE proposal. Establishment of a subgroup working on interface evaluation for the CEC, chaired by F. Drubay (next Session: June 1, 1978 in Brussels).

TC 6 (Man-Machine Communications):

The participation of active members was less than normal, this due to absence of three members representing their companies in other business. The main activity was allocated to work on new guidelines for Man-Machine Communication Design. The

The work plan was set up for the member activities to take place until the next meeting in September 1978, Brussels. A draft proposal for the guideline should be ready for presentation at the IPW's International Meeting at Purdue in October 1978.

The basic ideas of this work will be discussed in a Round Table Discussion on "Industrial Computer Systems and Man-Machine Communications" at the IFAC

Congress 78 in Helsinki, Finland.

A planned workshop on the same subject, proposed for September 1978, is at present postponed until December 1978.

The chairmanship of TC 6 is expected to change at a later date this year.

TC 7 (System Reliability, Safety and Security):

Highlights were presentations by Dr. Trauboth and Mr. Ludewig; another important topic was the preparation of the IFAC Panel.

TC 8 (Real Time Operating Systems):

The work of TC 8 during the Spring Meeting can be divided in three parts:

- Presentation of the main working paper of the TC, the "Up to Date Report"
- Cooperation with other TC's
- Development of the TC 8 operating system model described in the report

The round-table discussion of the TC 8 Report, which was preceeded by a presentation of this paper to the whole workshop made it possible to reach a major audience. The Report has been well accepted and it seems that its importance as a reference document has been recognized although it is not yet in its final state.

The work of our TC has also been acknowledged by a resolution which was accepted by the plenary.

This resolution asks the different TCs to consider the TC 8 Report and either to work in conformity to it or to state the differences and the reasons for such differences. Our TC has agreed, that such differences will lead to a re-evaluation of the corresponding parts of the Report by the TC. (Insert E-II).

The cooperation with other TCs, mainly with TCs 1, 2 and 3, was carried on and intensified and a new cooperation with EDISG was started. Several interesting contributions to our work were made by members of these TCs which joined our sessions. The dates of the next meetings of EDISG and TC 8 were fixed in a way that common sessions are possible.

The development of the main ongoing work of our TC focussed on the problems of process-management and exception handling. Both topics had been prepared in working papers from Arne Spiegelhauer and Stephan Goodenough respectively. The discussions and results will lead to two renewed working papers available at the next neeting. Someother, minor problems were also discussed and will be summarized in the minutes.

It was decided that the Up to Date Report should not be updated this time and that the accumulated additions and changes to the Report should lead to a new edition before the next International Meeting in Lafayette.

INSERT E-II

RESOLUTION REGARDING THE WORK OF THE REAL-TIME OPERATING SYSTEMS COMMITTEE

Purdue Europe agrees:

that it supports the activities of TC 8 on RT-OS in establishing guidelines as described in the up to date report I1;

that any other activities within Purdue Europe which are related to the topics contained in this report should take into account this report and

(i) work in conformity with this

(ii) agree with TC 8 appropriate steps to achieve conformity between this report and their activities

or

(iii) agree with TC 8 that there are sound reasons for non conformance

EDISG (European Distributed Intelligence Study Group):

Highlights: TC 48 of IEC (International Electrical Committee accepted proposal on Multicontroller Configuration in a CAMAC Crate. - Report from Communication Subgroup has been received and discussed. - Subgroups on Operating Systems and on Microprocessor Parallel Bus Systems have been re-established. - Cooperation with TC 5 on interface evaluation project.

Appendices E-II through E-VI present the available Minutes of the Technical Committee Meetings held at Zurich in connection with this Workshop Meeting.

3. ELECTIONS

The elections brought some change in TC-chairmanship as Peter Elzer, Harald Walze and Rudolf Lauber did not stand for re-election. We want to thank them for their work in Purdue Europe!

TC 3, TC 5 and TC 7 have therefore appointed new committee-chairmen. This is the current list of PE officers:

PURDUE EUROPE Chairman:

N. E. Malagardis, BNI IRIA, P.B. 105, F-78150 LeChesnay, France

TC 1: G. Heller, Fachhochschule fur Technik, Speyerer Strasse 4, D-6800 Mannheim, W.Germany

TC 2: V. Haase, Institut fur Angewandte Informatik und Formale Beschreibungsverfahren, Postfach 6380, D-7500 Karlsruhe 1, W.Germany

TC 3: D. Shorter, Corporate Engineering Lab., British Steel Corporation, 140 Battersea Park Rd., London SW 11 4LZ, United Kingdom

TC 4: H. Windauer, Math. Beratungs- und Programmierungsdienst GmbH, Glogauer Strasse 2a, D-2120 Lüneburg, W.Germany

TC 5: I. Hooton, A.E.R.E. Harwell, Oxford OX11 ORA, United Kingdom

TC 6: A. B. Aune, SINTEF, N-7034 Trondheim NTH, Norway

TC 7: J. R. Taylor, Risø National Laboratory, DK-4000 Roskilde, Denmark

TC 8: T. Lalive d'Epinay, Hybrid-Rechenzentrum der ETH, Voltastr. 18, CH-8044 Zurich, Switzerland

EDISG: K. -D. Müller, *ZEL/NE, Kernforschungsanlage*, Postfach 1913, D-5170 Jülich, W.Germany

4. SOME STATISTICS ON PURDUE EUROPE

The following data are based on the evaluation of a questionnaire which was answered by the Technical Committees of PE in winter 77/78. The figures are intended to show the scope of interest as well as the geographical (and political) distribution of the experts cooperating within Purdue Europe.

MEMBERSHIP

The nine TCs of PE (TC 1 - TC 8, EDISG) comprise approximately 200 active members, and approximately 200

corresponding or observing members. The range of active members per committee is from 10 to 50; the largest numbers can be claimed by TC 3, TC 7 and EDISG.

The geographical distribution shows experts from 17 European countries, in particular:

7 EC countries:	B, D, DK, F, GB, I, NL
5 Eastern European countries:	CS, H, PL, R. SU
5 Others	A, CH, N, S, SF

(Underlined countries are represented in all Technical Committees.)

The statistics of affiliations shows members from all relevant institutions engaged in industrial computer systems; research people are predominant:

Affiliations

University	30%
Research Centre	30%
Computer Manufacturer	15%
Industrial User	15%
System House Administration etc.	10%

ACTIVITIES

Purdue Europe was established in 1974 (TC 3, the LTPL-committee as a forerunner in 1971) and has held approximately 150 meetings of technical committees up to this time (most of them sponsored by the CEC). The technical results are to be found in approximately 650 technical

papers (among them approximately 400 from TC 3), and various publications in scientific journals, conference proceedings and research reports.

The current activities of the TCs of Purdue Europe show the following major concerns:

Language Standardization (together with national and international standardization bodies):

TC 1 (FORTRAN - ISA, ISO)

TC 2 (BASIC - ANSI, ECMA)

TC 3 (PLIP - ISO)

Interface Standardization (esp. in the scope of EC's Working Group of Standards):

TC 5 and EDISG

Cooperation with national and international research and development projects:

TC 3 (US-DoD: High Order Language Project)

TC 7 (Planned European Project on Safety and Security)

EDISG (British Project on Distributed Computing)

Evaluation of User requirements and preparation of guide-lines:

TC 4 (Software for Industrial Applications)

TC 6 (Man-Machine-Communications Design, Including Human Factors)

TC 7 (Safety and Security in Industrial Computer Systems)

Research Work; Observation, Classification and Enhancement of Industrial Computing:

TC 3 (Languages)

TC 8 (Operating systems)

All other committees are engaged in this work as well.

5. **PURDUE EUROPE AND THE CEC**

This is a copy of a letter from the Commission of the European Communities (DG III) to N. Malagardis, chairman of PE:

COMMISSION
OF THE
EUROPEAN COMMUNITIES

Directorate-General
for internal market
and industrial affairs

III/B/2

Brussels,....................
Kt / ad

Mr. MALAGARDIS
Bureau d'Orientation de la
Normalisation en Informatique
Domaine de Voluceau
Rocquencourt
B.P. 105
F 78150 LE CHESNAY

Dear Mr. Malagardis,

Purdue Europe and WGS.

In response to the spring 1977 resolution of support by Purdue Europe the working group on standardisation welcomes your offer of support. We would consequently like to forge more formal links between your technical committees and the WGS within the scope of work and procedures currently being agreed within that group. As you are aware the WGS performs a coordination activity and consequently it would seem advisable to establish an agreement of the interworking of the two groups.

In the meantime, we would like to confirm our request and ask for your agreement for two of your technical committees to assist the Commission:

- EPW/TC 3 to review of USDOD

  A review and subsequent report is requested on the suitability of the USDOD language work specifically in the field of Industrial Real Time computing applications.

- EPW/TC 5 support

  The support of an exploratory study on speeding up the production of standards, using point to point parallel interconnection as a guinea pig.

We look forward to forging a progressively more formal cooperation arrangement as we jointly explore the possibilities

Yours sincerely,

signed C. LAYTON

6. FUTURE MEETINGS

TC 1	August 23-25, 1978	Brussels
TC 2	Sept. 14-15, 1978	Brussels
TC 3	May 31-June 2, 1978	Brussels
	Sept. 6-8, 1978	Brussels
TC 4	? Sept. 1978	Brussels
TC 5	June 29-30, 1978	Paris
TC 6	Sept. 6-8, 1978	Brussels
	Nov. 29-Dec. 1, 1978	Brussels
TC 7	June 28-30, 1978	Brussels
	Sept. 20-22, 1978	Brussels
TC 8	June 26-28, 1978	Brussels
	Sept. 4-6, 1978	Munich
	Nov. 13-15, 1978	Brussels
EDISG	Nov. 15-17, 1978	Brussels

International Purdue Workshop

Oct. 9-12, 1978	West Lafayette, Ind. (USA)

Purdue Europe Spring Meeting

April 3-6, 1979	ETH-Zurich

APPENDIX E-I

LIST OF REGISTRANTS

THE 1978 EUROPEAN REGIONAL MEETING

INTERNATIONAL PURDUE WORKSHOP ON INDUSTRIAL COMPUTER SYSTEMS

Federal Institute of Technology (ETH)
Zurich, Switzerland

April 4-7, 1978

AB Atomenergi - Fack, S-61101 Nykoeping, SWEDEN

Mr. Per Gunnar Sjoelin

AEG-Telefunken, Buecklestrasse 1-5, D-7750 Konstanz, GERMANY

Dr. Dirk Kroenig

Dr. Michael Topschowsky, Hohenzollerndamm 150, Berlin 13

Atomic Energy Authority - Atomic Energy Post Office Reactor Dept., Computation Lab, Cairo, EGYPT

Dr. F. A. Mohammed

Atomic Energy Research Establishment (AERE HARWELL), GREAT BRITAIN

Dr. R. Gilbert, Bldg. 7.12, Didcot, OX11 ORA

Dr. John Patrick Scanlon, Computer Science and Systems Div., Bldg. 7, Oxon OX11 ORA

Mr. Ivor Hooton, Bldg. H7.12, Didcot, Oxford OX11 ORA

BNI - BP 105, F-78150, Le Chesnay, FRANCE

Mr. Nicolas Malagardis

British Steel Corporation - 140 Battersea Park Rd., London SW11 4LZ, ENGLAND

Mr. George Nicholas Brander

Mr. David Shorter

Brown, Boveri & CIE

Dipl. Hartmut Krome, Ladenburg, GERMANY

Dr. S. Vaid, Dept. EDM, CH-5401 Baden, SWITZERLAND

CAP - Sogeti Logiciel - 5 Rue Louis LeJeune, F-92130 Montrouge, FRANCE

Mlle. Helene Fallour

CNEN-CSN Casaccia - Via Anguillarese KM 1 + 300, Roma, ITALY

Mr. Sandro Bologna

Central Electricity Research Labs - Kelvin Ave., Leatherhead, Surrey, KT22 7SE, ENGLAND

Mr. Andrew Wells

Central Research Inst. for Physics - POB 49, H-1525 Budapest, HUNGARY

Dipl.-Ing. Janos Biri

Mr. Janos Szlanko

CERN - Division DD/HP, CH-1211 Geneve 23, SWITZERLAND

Dipl.El.-Ing. Pierre Baehler

Commonwealth of the European Communities - Bldg. WG RPG/421, Rue De La Loi 200, Bruxelles 1049, BELGIUM

Mr. Kenneth Thompson

Computing & Automation Inst. H.A. Sci. - Kende U. 13-17, H-1111 Budapest, HUNGARY

Dr. Ivan Bach

Mr. Elio De Agostino

Dr. Janos Gertler

Consiglio Nazionale Delle Ricerche - Corso Stati Uniti, Padova, ITALY

Mr. Gaetano Trainito

Digital Equipment Co. - Fountain House, Reading TG1 7QN, ENGLAND

Mr. Iain Smith

D DV WS SP312 Siemens AG - Hofmannstr. 51, D-8000 Muenchen, GERMANY

Dipl. Joachim Teller

Electricite De France - 6 Quai Watier, F-78400 Chatou, FRANCE

Ing. Rene Montmayeul

EPFL-DE-LCD - CH. De Bellerive 16, CH-1007 Lausanne, SWITZERLAND

Mr. Henri Roethlisberger

Fachhochschule Bielefeld - Wilhelm-Bertelsmann-Str. 10, D-48000 Bielefeld, GERMANY

Prof. Dr. Leberecht Frevert

Fachhochschule Fuer Technik Mannheim - Speyerer Str. 4, D-6800 Mannheim 1, GERMANY

Prof. Dr.-Ing. Guenther Heller

Fak. Physik, Universitaet Frieburg - Hermann Herder Str. 3, D-7800 Freiburg, GERMANY

Dr. Wolfgang Schupp

Fiat-DSQO - Corso Marconi 20, I-10125 Turin, ITALY

Mr. Vitorio Musso

Foxboro - Dept. 330, Foxboro, Mass. 02035, U.S.A.

Mr. Richard H. Caro

GEB. Sulzer AG, ABT 23/IE - Hegifeldstrasse 30, CH-8404 Winterthur, SWITZERLAND

Mr. Hansrudi Bernegger

GEC Computers Ltd. - Elstree Way, Borehamwood, Herts WD61RX ENGLAND

Mr. Alan F. Chalmers

Gesellschaft Fuer Reaktorsicherheit - Forschungsgelaende, D-8046 Garching, GERMANY

Dr. Ing. Wolfgang D. Ehrenberger

Dipl. Herbert Schueller

Hahn-Meintner-Institut GmBH - Glienickerstr. 100, 1000 Berlin 39, BERMANY

Dr. Guenter Wiesner

Hasler AG - Belpstrasse 23, CH-3000 Bern, SWITZERLAND

Dipl. Hilmar Gutfeldt

Hybridrechenzentrum Der ETH - Voltastr.18, CH-8044, Zurich, SWITZERLAND

Ms. Erika Doerig

Dipl.-Ing. Franz Kuster

Dr. Thierry LaLive d'Epinay

Dipl.-Ing. Peter Wegmann

I.C.I. Ltd. Corporate Laboratory - P.O. Box 11, The Heath, Runcorn, Cheshire WA7 4OE, ENGLAND

Mr. James Richard Halsall

IDAS GMBH - Holzheimer Strasse 96, D-6250 Limburg/Lahn, GERMANY

Dr. Axel Kappatsch

Industr. Inst. for Autom. & Measurements - Al. Jerozolimskie 202, PL-02-222 Warszawa, POLAND

Dr. Jan Bialasiewicz

Inst. F. Angew. Informatik, Universitaet - Postfach 6380, D-7500 Karlsruhe 1, GERMANY

Dr. Volkmar H. Haase

Inst. F. Elektr. Messtechnik Tu Wien - Gusshausstrasse 25, A-1040 Wien, AUSTRIA

Dipl.-Ing. Jochen Ludewig

Institut Fuer Informatik, TUM - Arcisstr. 21, D-8000 Muenchen, GERMANY

Dipl. Gerhard Kratzer

Dipl. Gerhard Schrott

Interautomation AG - Neumarkt, Postfach 365, CH-5200 Brugg, SWITZERLAND

Mr. Jacques D. Leibu

IRIA - Domaine De Voluceau, F-78150 Le Chesnay, FRANCE

Ing. Marc Kronental

Kent Automation Systems Ltd. - 6 Hunting Gate, Hitching, Herts, ENGLAND

Ms. Marion Jean Dearlove

Kernforschung Karlsruhe - Postfach 3640, 7500 Karlsruhe 1, GERMANY

Dipl.-Ing. Jochen Ludewig

KFA Juelich - KFA Juelich Postfach 1913, D-517 Juelich, GERMANY

Dipl.-Ing. Horst Walter Halling

Dr. Klaus Dieter Mueller

KFK/IDT - Kernforschungszentrum, 75 Karlsruhe, GERMANY

Prof. Heinz Trauboth

Dipl.-Ing. Harald Walze

Landis & Gyr - CH-6300 Zug, SWITZERLAND

Dr. Rudolf Schild

MBP, Math. Ber. U. Progr. Dienst - Glogauer Str. 2A, D-2120 Lueneburg, GERMANY

Dr. Hans Windauer

MERA-PIAP - Al. Jerozolimskie 202, PL-02-222 Warszawa, POLAND

Mr. Kazimierz Maliszewski

Dr. Henry Orlowski

Mission A L'Informatique - 24 Rue De L'Universite, F-75700 Paris, FRANCE

Mr. Dominique Bellanger

MOD(PE), R.S.R.E. - St. Andrews Road, Malvern, Worcs WR14 3PS, ENGLAND

Mr. Stephen John Goodenough

National Board for Techn. Dev. - P.O. Box S-100 72, Stockholm, SWEDEN

Dr. Bjoern T. Cronhjort

Norw. Inst. of Technology - Div. of Engineering Cybernetics, N-7034, Trondheim-NTH, NORWAY

Mr. Odd Pettersen

OEBB, GD/IK - Althanstrasse 2, A-1090 Wien, AUSTRIA

Dr. Robert Genser

Philips Data Systems - 4 Ave General LeClerc, F-92260, Fontenay-Aux-Roses, FRANCE

Ing. Jean Louis Lacour

Ing. Drago Vojnovic

Purdue University - 102 Michael Golden Labs., West Lafayette, INDIANA 47907 (USA)

Dr. Theodore J. Williams

Riso National Laboratory - 4000 Roskilde, DENMARK

Mr. John Robert Taylor

Sandoz - 4002 Basle, SWITZERLAND

Dr. Peter Fink

SCS - Oehleckerring 40, D-2000 Hamburg 62, GERMANY

Mr. Guenter Musstopf

SEPA S.P.A. - Lungo Stura Lazio 45, I-10156 Torino, ITALY

Mr. Carlo Santucci

Servolaboratory - Bygn. 327, DTH, DK-2800 Lungby, DENMARK

Mr. A. V. Spiegelhauer

SGAE - Lenaugasse 10, A-1082 Wien, AUSTRIA

Dr. Wolfgang Attwenger

Siemens AG, E STE 33 - Rheinbrueckenstr. 50, D-7500 Karlsruhe WEST GERMANY

Dr. Dipl. Horst Mittendorf

SINTEF Div. Automatic Control - N 7034 Trondheim-NTH, NORWAY

Dr. Ing. Arthur B. Aune

Systems Designers LD. - High Street, Frimley, Surrey 9016 5HJ, UNITED KINGDOM

Mr. William H. Hockey

Mr. Anthony Levene

Techn. Res. Centre of Finland - VTT/SAH, Otakaari 5I, SF-02150 Espoo 15, FINLAND

Dr. Pentti Tapio Uuspaeae

Tu Wien, Prozessrchenzentrum - Karlsplatz 13, A-1040 Wien, AUSTRIA

Dipl.-Ing. Werner Koblitz

Twente University of Technology - Postbus 217, Enschede, NETHERLANDS

Prof. Ir. Johannes E. Rijnsdorp

UKAEA - MPD, B521, AERE Harwell, Didcot, Oxfordshire OX11 ORA, ENGLAND

Dr. Martyn John Cooper

Mr. Alan Lewis

UKAEA Safety & Reliability Directorate - Wigshaw Lane Culcheth, Warrington WA3 4NE, ENGLAND

Mr. Andrew Aitken

University of Erlangen - Erwin Rommel Strasse 1, D-8520 Erlangen, GERMANY

Dipl. Peter F. Elzer

Univ. Karlsruhe Transp. U. Verkehrssyst - Kaiserstrasse 12, D-7500 Karlsruhe 1, GERMANY

Dipl. Karlheinz Kapp

Universitaet Stuttgart - Pfaffenwaldring 9, D-7000 Stuttgart 80, GERMANY

Dipl.-Ing. Rainer Kluttig

Prof. Dr. Rudolf J. Lauber

University of York - Heslington, York YO15 DD, ENGLAND

Mr. Terence J. Frogatt

Prof. Ian C. Pyle

U.S. Department of Defense - 1400 Wilson Blvd., Arlington, VA 22209 - USA

Lt. Col. William A. Whitaker

Weber AG - CH-6020 Emmenbruecke, SWITZERLAND

El.-Ing. Walter K. Spiess

Werkzeugmaschinenlabor, RWTH Aachen - Arnold Sommerfeldstr. 53B, D-5100 Aachen, GERMANY

Dipl.-Ing. York Tuechelmann

APPENDIX E-II

TC-2

REAL-TIME INDUSTRIAL BASIC COMMITTEE

International Purdue Workshop
Industrial Computer Systems

PURDUE LABORATORY FOR
APPLIED INDUSTRIAL CONTROL
102 Michael Golden
Purdue University
West Lafayette, Indiana 47907, USA
317/494-8425

Please reply to: Volkmar H. Haase
Institut für Angewandte Informatik
u. Formale Beschreibungsverfahren
Postfach 6380
D-7500 Karlsruhe 1
Telefon (0721) 608-3710
Telex 07-826-521

78-07-10

IRTB-E-10/78

Minutes

of the 14th meeting of TC-2 Purdue Europe

Participants

M. J. Dearlove	Kent Autom., Hitchin	GB
V. Haase	Univ. Karlsruhe	D
H. Halling	KFA Jülich	D
W. Koblitz	TU Wien	A
A. Lewis	AERE Harwell	GB
W. Schupp	Univ. Freiburg	D
J. Staeheli	Sandoz, Basel	CH
K. Stegner	Biochemie, Kundl	A
J. Szlanko	KFKI Budapest	H
N. Trainito	CNEN, Padova	I

Location

Eidgenössische Technische Hochschule, Zurich, CH

Date

April 4-6, 1978

Agenda:
1. RT-Enhancement to BASIC
2. Error-Control Enhancement to BASIC
3. Questionnaire Publication

0. General

This meeting was embedded into the annual regional meeting of Purdue Europe. Thus the time for technical work of TC-2's own interest was limited.

1. "Final: corrections in the RT-enhancement proposal were made, which shall be handed over to ANSI and ECMA in autumn this year. The result of the discussion is reflected in the latest version of the document (edited by A. Lewis).

2. J. Szlankō submitted proposals of both an Exception-handling and a Debugging Enhancement. They were discussed and will be updated. The Exception-handling document should become "stable" (in the ANSI-sense) in spring '79.

3. The publication of the Questionnaire (editor: J. Staeheli) was postponed. The topic will be raised again in the next meeting; possible an internal report for the CEC will be prepared.

Yours

COMMITTEE CORRESPONDENCE

american national standards committees:
X3—Computers & Information Processing
X4—Office Machines & Supplies
operating under the procedures of the American National Standards Institute

secretariat: CBEMA, 1828 L St NW (suite 1200), Washington DC 20036 202/466—2299

Doc. No. :

Date : February 22, 1

Project :

Milestone :

Reply to:

Volkmar Hasse, Chairman
European Branch, TC2
Institute F. Angewandte Informatik
Postfach 6380
D-7500 Karlsruhe 1
WEST GERMANY

Dear Dr. Hasse:

I would like to express my appreciation for the contributions TC2 (Durope) is making to the development of a standard for the programming language BASIC. The close collaboration involving TC2 (Durope), ECMA/TC21, and ANSI/X3J2 will ensure a viable international standard for BASIC in the shortest possible time.

Thank you for accepting responsibility for the Error Control Enhancement, in addition to the Real Time Enhancement. I might add that I was most impressed with the progress made on the Real Time Enhancement at the joint meeting of the three committees in London in November. We look forward to the next draft.

Very truly yours,

Thomas E. Kurtz

Thomas E. Kurtz
Chairman, X3J2

cs

APPENDIX E-III

TC-3

LONG TERM PROCEDURAL LANGUAGES COMMITTEE

L T P L - E U R O P E A N G R O U P	
Author: R. Gilbert & D. Shorter	LTPL-E/ RGDS 780531 (with amendments)*
Institution: Building 7.12, AERE., Harwell, Oxfordshire, OX11 ORA England	Category: M
	Updates: None
	Replaces: None
	Status:
Date (assigned): 5th April, 1978	Distribution: Those present
Date (completed): 31st May, 1978	pp: 12

Title: Minutes of the 41st LTPL-E Meeting, Zurich, (TC3 of the Purdue Europe Workshop) 4th - 7th April, 1978 (Draft)

Contents: If you have any comments or corrections could you send them to me to arrive not later than the 30th June (at the address given above).

R. Gilbert.

* This copy has been amended to take into account comments made at the 42nd LTPL-E meeting in Brussels.

1. Present

Smith, I.	Digital Equipment
Pyle, I.C.	University of York
Gertler, J.	Hungarian Academy of Sciences
Kappatsch, A.	IDAS GmbH
Mittendorf, [illegible]	Siemens
Williams, T.J.	Purdue University
Whitaker, W.A. (Part time)	US DoD
Teller, J.	Siemens
Froggatt, T.J.	University of York
Kronental, M.	IRIA/BNI
Malagardis, N.E.	IRIA/BNI
Shorter, D.N. (Chairman)	BSC
Gilbert, R. (Secretary)	AERE, Harwell
Chalmers, A.F.	GEC Computers Ltd.
Cronhjort, B.T.	National Swedish Board for Technical Development
Elzer, P. (Retiring Chairman)	University of Erlangen
Thompson, K (Part time)	CEC

2. Apologies for absence

Verroust, G.	IPN/Universite Paris-Sud
Holmes, G.W.	Systems Designers Ltd.
Barnes, J.G.P.	I.C.I.

3. Presentation and discussion of the evaluation of the US DOD Phase 1 results by the LTPL-E sub-group (Agenda item 3)

Peter Elzer had previously informed the workshop that the analysis had been performed by

P. Elzer	University of Erlangen	W. Germany
J. Harivel	CAP-Sogeti	France
J. Heger	B.B.C.	W. Germany
M. Minel	ECA-Autom.	France
K. Timmesfeld	IDAS	W. Germany
I. Wand	University of York	UK

Unfortunately R. Maddock had to withdraw.
Ratings were (best first) Green, Yellow, Red, Blue.

ELZER: We concentrated on parallelism, exception handling, I/O, algorithmic and descriptive facilities and used table format to show what was in the languages. After an initial disagreement on order, we soon agreed we did not like Blue, as being too messy and too complex. In fact, all languages were felt to be too complex, and to review them we had to take viewpoint of an educated system user (similar to position taken by LTPL-A). Green soon appeared to be best, although there was some uncertainty over whether the problems were really solved - e.g. we could not prove the 'box' method for synchronisation was sufficient although we knew the problems with other mechanisms. The style of Green was well liked.

At first Red looked more practical than Yellow, but on investigation it appeared that Yellow had identified problem areas but decided not to invent new mechanisms, rather to map onto functions. A majority of the team liked the independent tasking, and Yellow appeared easiest to implement.

Conclusion was that Green should be pursued further in any case and Yellow as safe back up solution.

WHITAKER: We had 80 analyses returned, of variable quality but generally pretty good.

The analyses were processed (with Peter Elzer's involvement) and two languages will go forward. The colour coding was Blue - Softech, Green - CSI-Honeywell-Bull, Red - Intermetrics and Yellow - SRI. Some changes must be made to the successful language proposals and to DOD requirements (hence Steelman). We are only 20% of the way to final language.

Jean Robert had written a paper on character sets, and as a result DOD might restrict the initial 64 character set further.

Steelman should be ready in 3 weeks for review by successful contractors. The language will be reviewed in the light of the analyses. In particular, there is a general feeling that the languages should be simpler. By April 1979 we will have completed the design and language manuals and have looked at implementation requirements. Then elaborate test procedures will follow leading to compilers and tools in 1980.

Another activity is now starting on programming environment, support aids, operating systems and tools. A Pebbleperson document should be ready on 21/4/78 covering these requirements. Some 80% of the total effort is seen to come from outside the actual language.

In the past, co-operation between US-DOD and Purdue Europe-LTPL has been extremely useful and fruitful. US-DOD have responded in deeds (rather than words). I want to thank the Committee, and hope co-operation will continue.

The analyses, manuals and the Steelman document will be sent to the Committee on microfiche, together with Peter Elzer's report on his participation in the review of the analyses and the sample problems generated by D.A. Fisher for the contractors.

SHORTER: How do we establish a European industrial view? Possibly via a meeting in Brussels with representatives of evaluation teams?

ELZER: The European evaluation teams were (a personal view)

European	-	LTPL - 1,2,3
W. Germany	-	IABG - 2,3
Finland	-	University of Tampere - 1?, 2
France	-	M. Parayre, CPM - 2,3
	-	Academic, AFCET - 2?
	-	M.J. Robert, CAP-Sogeti - 1,2,3
Netherlands ~~Norway~~	-	Prof. Dijkstra
Sweden	-	Defence team
UK	-	Academic
	-	DoI - 1,2,3
	-	MoD - 2,3
	-	RSRE

where 1 - denotes an industrial bias
2 - denotes should be invited to participate in generation of consensus

3 - denotes included LTPL members

WHITAKER: Having rejected unreliable analyses we found little difference between military and industrial responses. I also suggest you now look only at the two successful languages, and from these should evolve cleaned up versions in July. I have a list of people from Europe requesting information and this could help in setting up a forum.

A discussion then ensued on whether LTPL's functional requirements needed extension or whether they could proceed solely by evaluating the successful languages against examples - no final conclusion was reached.

SHORTER: We could have two meetings - one to establish a consensus and the second to present it to a wider forum.

ELZER: Could we present the view at IFAC?

GERTLER: Just one of 300 lectures. There could be an informal meeting..............

WILLIAMS: ....which should be publicised ahead.

PYLE: Publicity is necessary using all channels - the feedback element is most important.

SHORTER : The letter from the Commission is ready to go expressing the interest in establishing a consensus through LTPL-E.

SHORTER : We will know the winning languages before our meeting at the end of May. Before then we could distribute the European analyses.

General: Should the analyses be circulated to the evaluation teams or to the wider forum? Conclusion: circulate them as background information to evaluation team representatives who would participate in the Brussels meeting.

Further discussion was deferred until the following day.

4. Discussion of position papers on further work (Agenda item 4)

ELZER: There are three papers - G. Holmes (GWH 780206), P. Elzer (PE 780308) and A. Chalmers (AFC 780403). LTPL-E's new situation makes a discussion on future work items necessary. I agree with G. Holmes but also urge that we work on a booklet of test cases which we can use to judge new languages by coding the problems.

KRONENTAL/
SHORTER: There are examples in the 4 language proposals, and the DoI analysis.

General: Agreed to include work on test cases.

PYLE: We can reduce workload by co-operating with TC1 and 2 (on reviewing current practices), and TC8 (on operating systems).

ELZER: Mr. Wyngard from TC4 wants to work on hierarchies of description.............

PYLE: ....and this could link with look-ahead activities. LTPL-E should not leave this to TC4. In reviewing current practices "taking account of need to lead into a new language" is very significant.

There then followed a general discussion on desirable changes to AFC 780403. These were largely incorporated in AFC 780404 discussed later, but the following specific points were also made:-

TELLER: Inside ECMA there is a Real-Time Tasking Group chaired by Tony Beswick Ferranti, and including Bob Maddock (IBM), J. Ichbiah (CII), and AEG. Their activities are relevant.

KAPPATSCH: We should include non-procedural languages.

SHORTER: The "review of existing practices" is a bridge building exercise.

PYLE: Agree, and the transition problem is very significant - think of IBM and PL/1. Conversion aids are important.

WILLIAMS: Has there been any check on other LTPL committees? LTPL-A has agreed to act as an "industrial conscience". Why is LTPL-E's remit so complex?

CHALMERS/ SHORTER: No checking has been possible (time factor) but there is no conflict. The remit is complex because of need to spell out work following the CEC's difficulties in supporting the LTPL project, and the need to ensure LTPL-E and the CEC agree on remit.

ELZER: The section on LTPL-E organisation is somewhat personalised.

It was later agreed to omit this from the final document.

WILLIAMS: If we make the decision that "DoD-1" is the only likely language, then the remit should be rephased.

PYLE: We do not want to assume this now, but must prepare ourselves to make a decision when phase 2 is complete.

WHITAKER: I see the new remit as complementing our language work, and helping out with the Pebbleman exercise.

5. Election of new chairman (Agenda item 2)

Sinee Peter Elzer is leaving the University of Erlangen to work with the High Order Language Team at I.D.A. Arlington, he will be unable to continue as Chairman of LTPL-E. After a discussion, D.N. Shorter was proposed by I.C. Pyle and seconded by N. Malagardis. There were no other candidates, and D.N. Shorter was elected by those present with two abstentions and all other votes in favour

CHALMERS: I propose a vote of thanks to Peter Elzer for all his work on behalf of LTPL-E.

This proposal was passed unanimously.

6. Discussion of ACMKDS 780404 (Proposals for the way ahead for LTPL-E)

A draft of the paper ACMKDS 780404 was circulated for discussion.

THOMPSON: I have a comment about the introduction. The CEC hasn't rejected the LTPL-E Project Plan. I would rather say that, because there has been no agreement among member states, the CEC has not accepted the plan.

SHORTER: Let us look at paragraphs 5.1 to 5.4. How much work is left to be done in the IO area?

KAPPATSCH: There is still a lot to be done. We have to decide what the overall mechanisms will be and how they will be described. I estimate that there is at least 1 year's work.

KRONENTAL: The DoD IO work is not adequate. If we continue with the IO subgroup's work we may have some good recommendations. Paragraph 5.3 needs an extra subsection to cover this.

KAPPATSCH: This work should not go under the heading "Monitoring DoD environment work".

ELZER: I support the proposal that the IO subgroup completes its work. It is not clear whether the DoD will end up with user oriented IO or minimal (basic) IO. However, industrial users do need user oriented IO. There are therefore two possible customers for the work

- The DoD
- Industry.

THOMPSON: Are you covering human interface problems in your work? If so you need to establish a liaison with other Technical Committees.

KAPPATSCH: No, we are considering the language point of view and only look at man-machine aspects in so far as any language design work needs to.

SHORTER: The IO group should no longer be developing a language but instead be developing requirements.

ELZER: We must, as Mr. Thompson says, take other work into account, but we must complete existing work which I understand to be developing basic mechanisms. The IO group must produce a report to the same level as the tasking and algorithmic reports.

KAPPATSCH: It would take too long. Results in 1 year's time would be too late for the DoD work.

WILLIAMS: I feel that people are assuming that LTPL-E will talk to DoD direct. In fact, LTPL-E should communicate with DoD via LTPL-C. In the past this has not happened because of the need to respond very quickly.

SHORTER: I accept that in the ideal situation our results should go through LTPL-C. However, I don't think we can do that in this case. There may be time to send a draft of the IO proposals through LTPL-C to DoD but the final results would have to go direct to DoD.

KAPPATSCH: One year from now Phase II will have finished and the IO proposals will be of no use to DoD.

FROGGATT: Why not just consolidate your past work? Produce a paper saying what has been done and what is left.

KAPPATSCH: That is much less work and would only take one meeting.

SHORTER: We should consolidate our work and tell DoD that we are prepared to react to their proposals. We should defer further work on IO until we see what the DoD language is like.

ELZER: Our work should be in the form of guidelines which can be used to evaluate the DoD I.O. proposals when they appear. The guidelines should be a cross between a functional specification and a checklist.

KRONENTAL: In ISO we are producing a set of criteria for real-time languages and it includes I.O.

SHORTER: We now have a course of action:

1. Produce our consolidated report (at our next meeting).

2. Work on language mechanisms in the contect of the two successful DoD languages (about 6 months) and present the results at the next International Purdue Workshop.

3. Get information from existing checklist (e.g. ISO).

Within a year we should have a set of documents which will be useful for measuring the DoD proposals against industrial user requirements.

SMITH: What are you using as a definition of industrial user requirements?

WILLIAMS: Like the Green Sheets

SHORTER: The Green Sheets need updating because they are out of date and incomplete.

WILLIAMS: There's a job for someone.

ELZER: It has been difficult in the past to revive a group on functional requirements.

THOMPSON: Discussing functional requirements is a good way to end up doing nothing.

One possible activity is to act as the spearhead of other groups (TC1, TC2, DoD) to bring them together and ensure unification.

SHORTER: The chairman of the I.O. subgroup should prepare a paper for the next meeting in Brussels giving the new remit for the I.O. subgroup.

The meeting went on to discuss the production of example problems.

SHORTER: Mr. Whitaker has said that it can be misleading if we only have small examples.

ELZER: We should start with small examples and develop them into larger ones.

The following people agreed to form an Examples Subgroup:

Kronental (Chairman)

Mr. Kronental agreed to write to the A list to ask for sample problems and to say that a discussion will be held at the next LTPL-E meeting.

The meeting went on to discuss setting up a meeting to review the DoD evaluations.

THOMPSON: The Commission has formally requested LTPL-E to provide an industrial view of the impact of the DoD HOL project within Europe and to report on DoD's progress. The request was timed so that LTPL-E could take part in the analysis but this was not part of the request.

It was suggested within LTPL-E that it would be useful to see what other groups had thought about the DoD by setting up a post-analysis meeting.

I must stress the need to communicate directly with member states. If a problem arises in the future which requires action from the Commission, you must first generate political awareness in the member states. The Commission will then be able to proceed because the member states will be ready to act.

TELLER: The language evaluation is in the past, we should wait for Steelman.

THOMPSON: In setting up the meeting you should be careful to involve only those people who will put forward an industrial view.

SHORTER: At the next LTPL-E meeting we need a subgroup to discuss the evaluation and to draft a report which will first go to the full LTPL-E group and then to the Commission. The subgroup will only include those members of LTPL who were involved with the evaluation plus people from other evaluation groups.

WILLIAMS: The report should go to a group of users before going to the Commission.

ELZER: The report could be distributed as broadly as possible with a request for comments.

THOMPSON: If you distribute the paper you must be prepared to process the replies.

SHORTER: We now have the following course of action:

1. The Evaluation Subgroup drafts a paper.
2. The full LTPL-E group reviews it.
3. We get user contributions by
   (a) setting up a user group (1 day?)
   (b) distributing the paper.
4. Finally the paper goes to the Commission.

It was agreed that the paper should also go to other groups such as LTPL-A, LTPL-J, ECMA.

The meeting then discussed setting up a subgroup to consider the industrial requirements for development and running aids. The following people agreed to serve on the subgroup:

Chalmers (Chairman)
Shorter
Gilbert
Froggatt
Smith

They are to meet in Brussels and consider the industrial requirements in readiness for Pebbleman.

KAPPATSCH: Do the reliability, security and safety aspects of a higher order language comprise a useful topic?

SHORTER: It will be useful for you to set up a joint meeting with Dr. Taylor (the chairman of the Software Subgroup of the Safety and Security Committee). I will talk to Prof. Lauber.

Are there any other comments on ACMKDS 780404?

THOMPSON: There should be an extra paragraph (4.6) to mention the liaison with ISO, PLIP and the national standards bodies.

CHALMERS: I will distribute the corrected paper to the A list.

7. Paper list and Mailing list

SHORTER: Mr. Elzer has given me the current mailing lists and I will try to set up mechanisms for producing labels.

The latest paper list was given to Mr. Gilbert to distribute to the A list with the minutes of the previous meeting.

8. Next meetings

LTPL-E meeting, Brussels:	31st May - 2nd June
LTPL-E meeting, Brussels:	6th September - 8th September
International Purdue Workshop:	9th October - 13th October

9. The resolution of the Real-time Operating Systems Committee (TC8)

A resolution had been passed by the Purdue Europe Spring Meeting that required all technical committees to resolve any differences between their own work and that of TC8.

SHORTER: Do we need to go back and review the tasking paper in the light of the RTOS resolution?

FROGGATT: INC and DEC can be used to implement all tasking primitives. You have the logical ability but not an efficient implementation.

PYLE: The resolution did not expect us to accept the details of the TC 8 report, just to take it into account. We can say that, because of efficiency, we have different primitives at a particular level.

SHORTER: We should map our tasking primitives onto the various levels of the RTOS proposals. I will write to Mr. Timmesfeld to ask if this can be done.

When we know the two DoD languages someone should check their operating system requirements against the TC8 proposals.

PYLE: We should recommend that TC8 look at the DoD languages.

SHORTER: Perhaps our job is to extract the tasking elements of DoD and submit them to TC8.

PYLE: I am unhappy about TC8's comments on exception handling. TC8's paper is totally inadequate in this area. Neither do they say anything about scheduling strategy (i.e. how to choose what task to run). This is the main difference between TC1 and TC8. An operating system is only a <u>real time</u> operating system when the scheduling strategies are included.

SHORTER: Returning to exception handling, it is a topic that LTPL-E has not covered adequately. We don't have a model that includes exception handling.

KRONENTAL: The reason is that a simple model does not have an interpreter and without an interpreter exception handling cannot be included.

PYLE: There are two papers referenced by the documentation of the Green language.

- LAMPSON, MITCHELL, and SATTERTHWAITHE. On the transfer of control between contexts, Lecture notes in computer science, Vol. 19, Robinet (Ed), (Springer Verlag, New York 1974), pp 181-203.

- BRON, FOKKINGA, and DE HAAS. A proposal for dealing with abnormal termination of programs, Twenta University of Technology, Mem. No. 150, (Netherlands, Nov. 1974).

These two papers look at exception handling in two ways:

- Exceptions occur because something is wrong with the data
- Exceptions occur because something is wrong with the activity.

We should look at both papers and see which fits our own ideas.

There followed a brief discussion about exception handling. Mr. Froggatt agreed to produce a paper setting out the aspects of exception handling that need discussion within the LTPL-E group.

10. Any other business

SHORTER: We are to decide next time whether to ask someone from the Safety and Security Committee to talk to us about specification languages.

We have been invited by the Safety and Security Committee to send up to 4 people to their next meeting in September.

KRONENTAL: I have a message from Mr. Ichbiah to say that if CII Honeywell Bull get the contract from DoD they would like to have a close co-operation with LTPL especially with respect to the review and simplification of the language.

We can co-operate with DoD either by

- direct interaction with DoD
- interaction with the contractors

However, there are difficulties if we, as an ad-hoc group, are influencing the contractors, particularly if we interact with only one contractor.

GERTLER: If we work with only one of the two contractors we will not be asked to take part in the Phase II evaluation.

SHORTER: Any co-operation with one contractor should be extended to the other contractor. We should state what we want to do, let Whitaker know and invite his comments. I will write saying that we have been approached by one potential contractor and that we are willing to co-operate but want to maintain our independence so that we can give, and can be seen to give, impartial advice to the DoD and the CEC. I will go on to say that we intend to offer our co-operation to both contractors but would welcome DoD's views before doing so.

We must clear our channels of communication with national senior officials. I would like to see someone at each LTPL meeting who can report to LTPL the views of his own national senior officials and, in return, report to the national officials the views of LTPL. In this way I want to avoid the situation where we appear to have agreement with national officials but, in fact, have none.

* KRONENTAL: Perhaps we should write a short note summarising LTPL's view and this note can be sent to a special mailing list after each LTPL meeting.

* SHORTER: We could set up a small subgroup to draft this paper and whose members could report back.

* KRONENTAL: I have seen newspaper articles written about LTPL by people who have no contact with LTPL. Perhaps we should write our own articles.

* SHORTER: This is something else that could be covered by the same subgroup.

Meeting closed.

* The last four items in these minutes are to be clarified in the minutes of the 42nd LTPL-E meeting, Brussels, 31st May - 2nd June.

APPENDIX E-IV

TC-4

PROBLEM ORIENTED LANGUAGES COMMITTEE

PURDUE EUROPE

TC 4

Technical Committee on

Problem Oriented Languages

<u>ANNUAL REPORT 1977 - 78</u>

Since the last spring meeting of PE, TC 4 met 2-times in Brussels with 11 and 13 members.

The main topic of our work was to answer the questions "what is a POL?" and "what is the direction of our work?"

In the past a POL, a Problem Oriented Language, was considered only as a special application system, that means a software package together with a command language in form of sentences or fill-in-the-blanks-forms.

The user of such a system should not have to be a programmer, he only has to give actual parameters to the package, which already contains the solution of his problem. But we have seen that special application systems usually don't fit special applications. They must be adapted or extended. Additionally some main program must be constructed to combine the various modules of the package with its specific parts. Today this is done by specialists using assembler - a very expensive and not portable way.

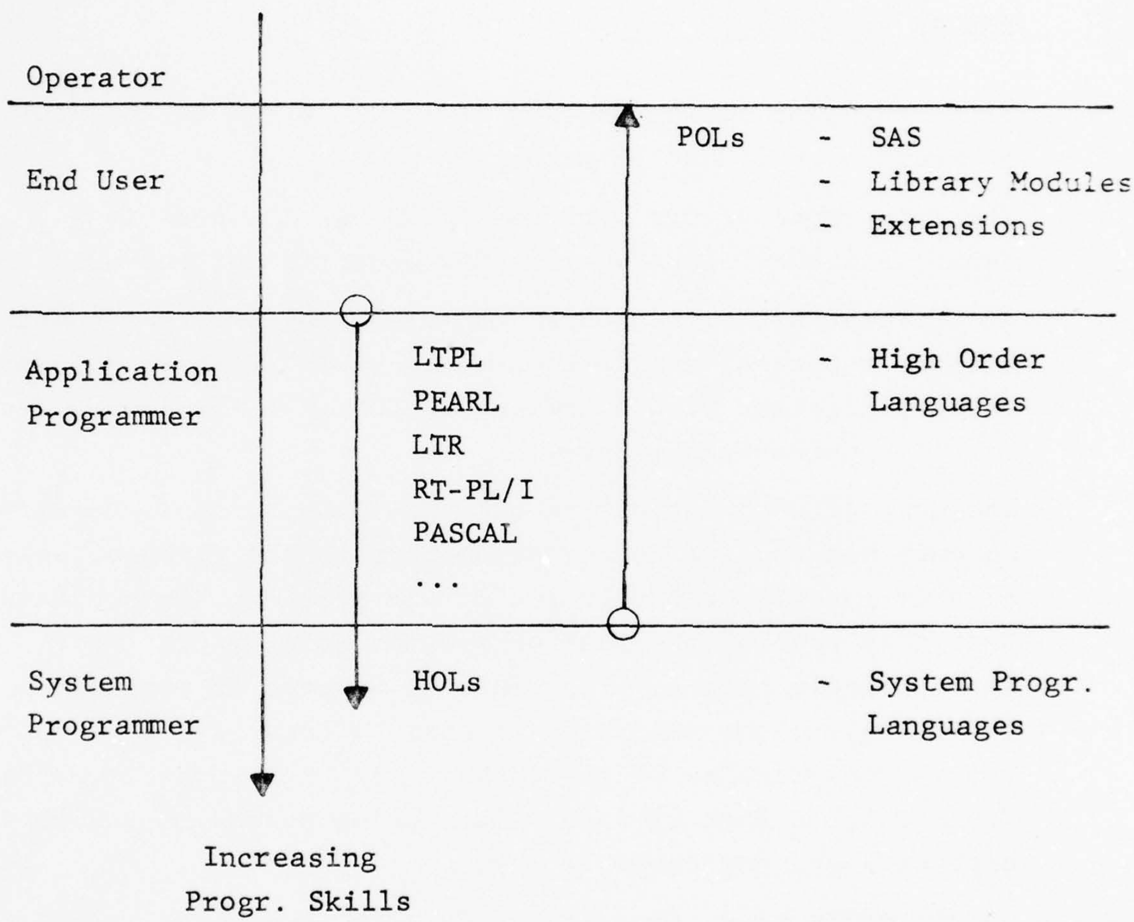
On the other hand, some of the new high order realtime languages like C-PASCAL or PEARL or Realtime-PL/I allow the engineer to program himself on a higher level than in the past without being a software specialist but having some programming skills.

So, TC 4 has formulated the following answer to the question "what is a POL?"

POLs can reach from special application systems (SAS), which contain the solution of a user's (engineer's) problem, to high order languages (HOL), which enable a user to program the solution himself on a very high level.

This situation is described by the following picture:

What is a POL?



The current activity of TC 4 is to analyze the user requirements on POLs in two ways:

1. What are the user requirements on POLs as special application systems?
   (Top down approach)

2. What are the requirements on POLs as high order languages, which enable the user to program individual (prototype) software and to construct flexible modules of application systems, which are reusable in classes of applications?
   (Bottom up approach)

The two results of this analysis have been written down in two corresponding working papers, which are still in the state of discussion. They will be finished during this year. However, especially the paper about POLs as very high order languages should be discussed with LTPL too.

In parallel we have started to look for POLs developed with the use of a high order language. At the last meeting we had a presentation about

> Experiences with the development of a structured program package and a POL using PEARL.

And at this Zurich meeting there will take place a presentation about

> Experiences with RT-FORTRAN using it to develop a software package for DNC.

H. Windauer

APPENDIX E-V

TC-5

INTERFACES AND DATA TRANSMISSION COMMITTEE

Zurich, April 4th, 1978

PURDUE EUROPE
T.C.5 INTERFACES AND DATA TRANSMISSION COMMITTEE

MINUTES OF THE 13th MEETING AT THE MAIN BUILDING OF THE E.T.H. IN ZÜRICH ON APRIL 4th, 1978 (14.00 - 18.00 hours)

The meeting was attended by:

H. Walze (Chairman)
W. Attwenger
H. Bernegger
F. Biri
J.R. Halsall
I.N. Hooton
R. Kluttig
W. Mohr
K.D. Müller

Address list is added as Appendix 1.

Those present wish to thank Dr. Th. Lalive d'Epinay and his colleagues for the excellent organization of the Purdue European Regional Meeting 1978.

The agenda previously circulated by the Chairman on March 20, 1978 comprised the following items:

1 Chairman Nomination

2 Acceptance of last minutes

3 Reports on recent meetings of related committees

4 Presentation, discussion, possible revision and adoption of new TC-5-System-Draft (SIRE)

5 TC-5-Activities carried out by request of CEC

6 Miscellaneous

7 Next meetings

During the preliminary discussion it was agreed that a new Agenda Item should be added after item 4 "REPLY TO EDISG" and that the items 5 to 7 should be shifted by one to items 6 to 8.

1. CHAIRMAN NOMINATION

Mr. Walze pointed out that the two designated chairmen Mr. B. Van den Dolder and Mr. F. Drubay will not attend the workshop.

It was proposed that Mr. I.N. Hooton should take over the chairmanship. Mr. I.N. Hooton accepted the nomination as chairman provided that his management will agree and that meetings would be arranged in Bruxelles (paid by the CEC), in England or in such a way that they will only be a small burden on the Harwell travel budget.

Mr. H. Walze acted as chairman for the 13th meeting.

2. ACCEPTANCE OF THE LAST MINUTES

Revised minutes dated 28/3/78 have been distributed by Mr.J.R. Halsall.

The following typing errors were detected:

Page 3: 3.3 Chairman Meeting 7th December 1977

Page 6 and first page of Appendix I: 4. Response from EDISG Communication Sub-Group to Sir.
The same typing error is three times in paragraph 4, one time in paragraph 7 and in the heading of Appendix I.

Second page of Appendix I (second line): .... that the connection to levels 2 and ......

Further comments:

Page 4: Statement 6.5 is not a characteristic of the System chosen, but a remark only. Therefore the number 6.5 should be deleted, the text should be taken as a remark.

Page 5: Last but one paragraph: starting with: "The chairman proposed a new compromise ...". Mr. H. Walze pointed out that one should carefully distinguish between the System structure and the recommended protocols. Therefore several possible protocols should be described in a separate chapter and should be examples only.

3. REPORTS ON RECENT MEETINGS OF RELATED COMMITTEES

EDISG:

EDISG had a full meeting in Bruxelles, March 8-10.
A new subgroup was formed which will look into the problems of Multi-Micro-Bussystems (parallel bus systems). Evaluation Criteria will not be comparable with the Functional Requirements for Proway (Process Data Highways for Distributed Process Control System) of IEC SC65A/WG6.
The Communication Subgroup of EDISG decided to establish a new questionaire which will be used to examine 17 communication subsystems (bit serial systems) which are in operation or have been proposed.

SC65A/WG6:

Besides the work done on the functional requirements two subgroups have been busy since the last full meeting (Vienna, October 1977) and provided a format for the presentation of candidate systems and a format for the comparison of candidate systems. TC5 Purdue America (Meeting held at IBM, Boca Raton, January 12-13, 1978) decided to input directly to the activities of WG6 on the format questions rather than support a separate parallel activity.

In the report from Mr. R.W.Gellie on the "Mission of TC5 (Purdue America) Summary of Responses" three pages deal with the Format for the Proposal of Candidate Systems to IEC SC65A/ WG6.

TC5 (Purdue America):

Members were requested to submit in confidence their personal views in three areas:

1. Future directions of the committee's activities
2. Whether or not an HDLC-based protocol should form the basis of the control procedures for the process data highway
3. Questions which, if answered by Dr. Funk, would enable the committee to evaluate the correctness and significance of his analysis.

PDV:

A well attended meeting on a bussystem for Machine Tool Control (MPST, parallel bus) was held in Stuttgart, February 23, 1978.

Mr. Walze reported, that a Japanese guidline for the "Evaluation and Comparison for Bit Serial Systems" has been prepared.

4. <u>PRESENTATION, DISCUSSION, POSSIBLE REVISION AND ADOPTION OF NEW TC 5-SYSTEM-DRAFT (SIRE)</u>

Mr. I.N. Hooton presented the foils he had prepared for his presentation which was scheduled for Thursday, April 6th afternoon.
During the presentation was a short discussion on the terminology used in the SIRE paper.
Furthermore it became evident that the term "fixed length" has to be clearly defined and that there must be discussions on the topic on which levels the watch-dog-mechanisms are to be implemented.
There is still a lot of work to be done defining various interfaces. Mr. I.N. Hooton is prepared to write down his own ideas as a basis for further discussions.

5. <u>REPLY TO EDISG</u>

This item was postponed and it was proposed to discuss it during an EDISG meeting at which TC5 members will be present as guests and which was scheduled for Wednesday, April 5th, afternoon.

6. TC 5 ACTIVITIES CARRIED OUT BY REQUEST OF CEC

Mr. I.N. Hooton gave a short description of the technical background of the work to be done.
The task of this feasibility study (analysis of the problem) is to find out, if a flexible point to point interface with an intermediate language between two standard stations could be a helpful tool to connect two separated busses (parallel busses) of the same type (first step) and of different types (second step).
There is a proposal ISO 7 which could serve as the possible intermediate interface.
Mr. K. Thompson (CEC) told the secretary of this TC5 meeting after the meeting that it is the intention of the CEC to place the contract with IRIA (Institut de Recherche d'Informatique et d'Automatique)- France - under the technical responsibility of BNI (Bureau d'Orientation de la Normalisation en Informatique).

7. MISCELLANEOUS

Topics for further meeting were briefly discussed. They are covered in paragraph 4 of these minutes.

8. NEXT MEETING

As outlined in the minutes of the 12th meeting the next TC5 meeting is planned to take place at IRIA (near Versailles) on the 29th and 30th June 1978.
Mr. I.N. Hooton was asked to contact his management with respect to his chairmanship as soon as possible and to get at least the permission to chair the next meeting.

W. ATTWENGER

APPENDIX E-VI

TC-7

RELIABILITY, SAFETY AND SECURITY COMMITTEE

Minutes of the 16th meeting of Purdue TC7
on safety and security

1. Guests:

Mr. Musso, observer from EC, working with documentation.

Mr. Ludewig, working with Dr. Trauboth, computer aided software specification and design.

2. Apologies from many.

Amendment to agenda. A topic on the Ispra cooperation was added to the agenda.

3. IFAC Panel - Helsinki 1978

Mr. Hendry is proposed as a replacement for Mr. Levene.

At Helsinki there will be a discussion panel to discuss:

a) An introductory presentation of the problems of safety related software.

b) A number of "prepared questions".

c) Questions from the audience.

4. Cooperation with CSNI

A letter had been received from the chairman of CSNI, proposing cooperation, more specifically organizing a symposium on methods for improving safety of computer based control systems.

We should make a plan, and forward it to CSNI and IFAC. The meeting should preferably be in 1979 if this can be achieved, but this may be difficult because of the IFAC time table.

A possible venue might be Munich or Stuttgart in September 78. A list of topics is appended to these minutes. Professor Lauber agreed to pursue the problem of arranging the meeting.

5. Election of Chairman

Mr. Taylor was elected as TC7 chairman.

Dr. Ehrenberger was elected as chairman of the software sub-committee.

A vote of thanks was recorded for Pr. Lauber, and his work for TC7 during the previous four years.

6. Next Meeting

June 28-30 in Brussels.

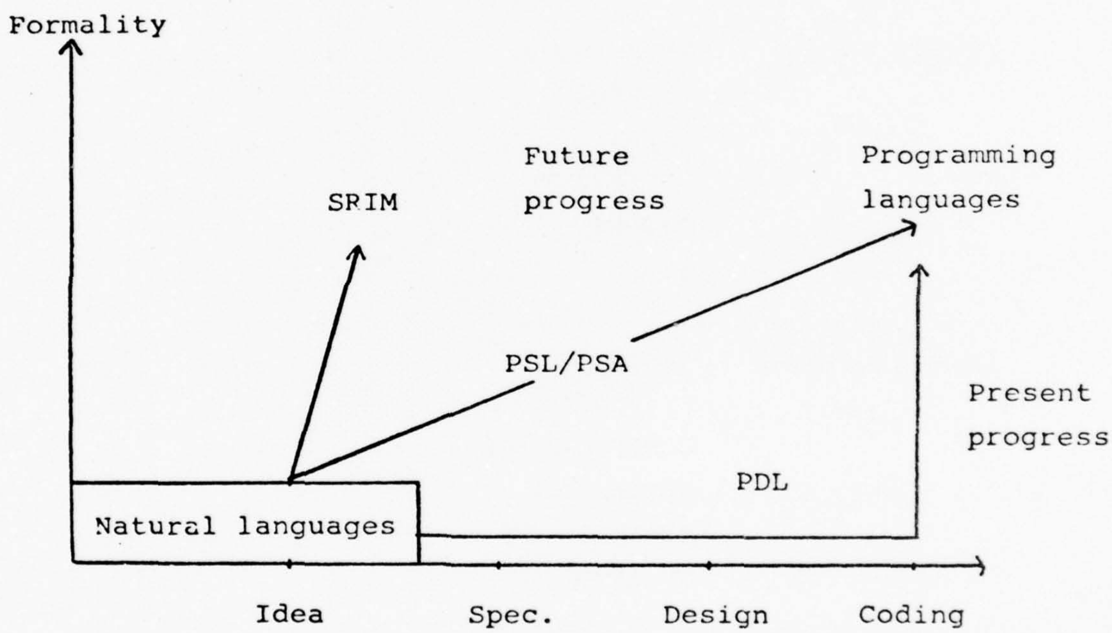
7. Methods and Tools for Software Specification and Design Mr. Ludewig

For a description of the presentation see working paper 149.

Specification	A precise statement of requirements that a product must satisfy (Parnas).
Abstract program	Description of a problem solution which does not contain details.

Procedural	Describes sequence of execution = sequence of statements.
Non procedural	Sequence of statements not relevant for execution.

Non procedural languages stress the static or invariant aspects of a system. Procedural stress the dynamic.



The process of development can be described by the diagram above, indication a progress towards more formal descriptions at earlier development stages.

<u>Question</u>: Isn't the amount of stored information the same, whether you use formal or informal natural language documentation?

<u>Answer</u>: The tools <u>force</u> you to provide good documentation. It is possible to develop good systems without such aids.

8. <u>PSL/PSA Features, Trauboth</u>

PSL/PSA is a notation for system description.

- Structured description of info systems.
- Storage of specifications.
- Partially suited for other development phases.
- Replaces or supplements manual documentation.
- Interactive selective output of design characteristics.

<u>Object types</u> (examples)

PROCESS
INPUT
INTERFACE
OUTPUT
SET

<u>Relationships</u> (examples)

RECEIVES
GENERATED BY
GENERATES
RECEIVED BY
UPDATES

This gives the <u>structure</u> of information processing.

From this can be extracted:

System static structure
Data flow
Data structure
Data origin and use cross references.

The system can also describe how the system behaves over time - when things happen and under what conditions.

Question: At which project size does this system become useful?

Answer: It depends on complexity. The cost of a report costs - 5 mark. 20 or 30 marks to input a problem. - Not for half a man year. - probably 5 man years. As a documentation tool from 3-6 man months - for larger projects as an analysis tool.

Draft List of Topics
Joint Symposium on Software for Safety Related Systems

Specification, design, and structuring of software for safety

- Design methods, design aids, specification methods.
- Redundant programming, fault tolerant software.
- Influence of software reliability on system reliability.

Project management for safety

- Quality control in software development.
- Programming guidelines.
- Program change control systems.

Testing, verifying, and licensing software

- Testing theory, automated test aids.
- Statistical testing, verification by simulation, statistics theory of S/W rel.
- Systematic testing "correctness proofs".
- Licensing criteria.

Applications with safety related software systems

- Railway, aircraft, traffic control systems.
- Nuclear, chemical, oil drilling and production.
- Mining, elevator, steel making.
- Reliability experience, reliability growth.

Documentation and documentation systems

- For safety.
- For licensing.
- Figures of merit and quantification of safety.

Testing and verification of real time multitask systems and operating systems

- Validation of multitask systems.
- Influence of operating systems.
- Verification of real time systems.
- Operating system design for safety.

9. Discussion of Proposed Ispra Project

Dr. Fangmeyer, Ispra, has proposed to organise a project for a simulator for microprocessor and software systems, which would allow failure simulation of microprocessor systems. The project would run in four phases:

- Feasibility study.
- Detailed design.
- Implementation.
- Validation and use.

TC7 would be invited to help with this work, the help possibly taking the form of contract work, the contract being let by the EC.

A discussion of the project was held.

Is the project feasible? - Mr. Santucci, who has considerable experience in this area, fealt that the requirement was extreme, but that the goals could be achieved.

A major problem is ensuring that the simulator is sufficiently fast - most simulators are very time consuming. On a Siemens 32, simulation of a single instruction takes 1 sec.

Intermixed multilevel simulation is necessary - this may mean that single bit representation of signals is undesirable - whole register contents, and integer or floating point representations for software may be required.

The system should be flexible and portable - indicating a FORTRAN implementation but special hardware might increase speed - can a multiprocessor be used?

The simulator should simulate the computer's environment, as well as its working.

The problem of ensuring a complete set of simulations should be treated.

There is little point in developing a new simulator (many exist already) unless it proves impossible to extend an existing simulator.

One might use an environment simulator to provide inputs for testing a real computer.

If the proposed change in status of Purdue Europe takes place, TC7 could contract to undertake the work. Otherwise, we could offer to provide a discussion forum, and committee member organisations could contract to carry out necessary work.

The project should pick out a number of applications at the start, sufficiently diverse to guarantee a flexible simulator design. Some TC7 member organisations would be interested in making use of such a simulator.

<u>Documentation Subgroup Minutes</u>

<u>Status of work</u>

Prologue	WP 118 list of work to be done. WP 135 scope part DRD-E-26. WP 138 documentation matrix. WP 141 documentation for acceptance.
Part 1	Documentation overview, part of DRD-E-26.
Part 2	System requirements, major part of DRD-E-26 (complete, subject to discussion).
Part 3	System description, internal subsubgroup papers.
Part 4	Technical support documentation, internal subgroup papers.
Part 5	Project management documentation, paper from PM.
Part 6	Maintenance documentation, <u>no progress</u>.
Part 7	Associated equipment and software, <u>no progress</u>.

Written comments were invited to part 2, to be adressed to Mr. cooper and Mr. Winter.

A vote of thanks was recorded for Mr. Winter's efforts and impressive results.

Paper 141 was presented by Dr. Genser. It provided a checklist of concepts to be taken into account in a standard for safety system documentation.

The subsubgroup on section 2 might take on the job of drafting a prologue.

Further copies of DRD-E-26 should be obtained from J.R.Taylor.

Part 3 is in four sections:

Section 1 - The system ...... status
some work completed,
1.1 and 1.2

Section 2 - The hardware ......

Section 3 - The software ......

Section 4 - The operator/user

Section 1 of part 3 has 5 chapters, and will probably be completed by September. We need expertise on hardware. We should obtain a copy of the TC6 standard on documentation.

SECTION II

MINUTES OF THE JAPANESE REGIONAL MEETING INTERNATIONAL PURDUE WORKSHOP ON INDUSTRIAL COMPUTER SYSTEMS

MINUTES OF THE JAPANESE REGIONAL MEETING

INTERNATIONAL PURDUE WORKSHOP ON INDUSTRIAL COMPUTER SYSTEMS

Japan Electronic Industry Development Association (JEIDA)

Tokyo, Japan

June 29-30, 1978

ORGANIZATION

The Fifth Japanese Regional Meeting of the International Purdue Workshop was held under the sponsorship of JEIDA (Japan Electronic Industry Development Association) on June 29-30, 1978, in Tokyo, Japan. There were one hundred and forty three (143) people in attendance at the Tokyo Workshop, representing both vendors and users of industrial computers (Appendix J-1). The Agenda of Insert J-1 was followed.

As has been the case in past years, the representatives of the Japanese Regional Meeting will present a report of their meeting at the International Workshop Meeting at Purdue University on October 9-12, 1978. This report will be carried in the Minutes of that meeting and will serve to give the details of the meeting listed above.

INSERT J-I

AGENDA

FIFTH JAPANESE REGIONAL MEETING

INTERNATIONAL PURDUE WORKSHOP ON INDUSTRIAL

COMPUTER SYSTEMS

Japan Electronic Industry Development Association (JEIDA)

Tokyo, Japan

June 29-30, 1978

THURSDAY, JUNE 29, 1978

10:00 - 10:30 Introductions of Workshop Program and Survey of JEIDA Activities on Industrial Computer Systems in 1977 (April 1977 - March 1978)

Dr. Kohei Sato
Chairman of Technology of Industrial Computer System Committee

10:30 - 12:00 Tutorial Lecture

Application of Industrial Computer Systems in the Machine Industry

Dr. Munetaka Jyotaki
Sumitomo Heavy Industries, Ltd.

12:00 - 13:00 Lunch

13:00 - 13:40 The Present Status of the DOD High Order Languages Project

Mr. Koichi Mori
Chairman of Software of Industrial Computer System

13:40 - 14:20 The Actual State of Software for the User

Mr. Ichiyo Shirai
Panafacom Ltd.

PRECEDING PAGE BLANK - NOT FILMED

INSERT J-I (Cont.)

14:20 - 15:00	Technical Trends of Operating Systems Mr. Shigeru Yamamoto Yokogawa Electric Works, Ltd.
15:00 - 15:10	Coffee Break
15:10 - 16:00	Installation Guidelines for Industrial Computer Systems Mr. Genji Fujimoro Hokushin Electric Works, Ltd.
16:00 - 17:00	The Present Status of Maintenance for the User

FRIDAY, JUNE 30, 1978

10:00 - 11:00	Technical Trends of RT-BASIC and Microprocessors Mr. Koji Yada Chairman of Industrial Microcomputer Committee
11:00 - 12:00	Technical Trends of Industrial Dataways (IEC/TC65A/WG6) Mr. Takashi Tohyama Chiyoda Chemical Engineering and Construction Co., Ltd.
12:00 - 13:00	Lunch
13:00 - 14:00	32 Bit Architecture for Industrial Computers Mr. Washei Yamanaka Tokyo Shibaura Electric Co., Ltd.
14:00 - 15:00	Practices in Data Bases for Distributed Industrial Systems Mr. Tatsuya Mutoh Mitsubishi Electric Corp.
15:00 - 15:10	Coffee Break

INSERT J-I (Cont.)

15:10 - 16:00 Standardization of CRT Operator's Consoles

Mr. Yoshio Tomita
Yokogawa Electric Works, Ltd.

16:00 - 17:00 Standardization of Software Descriptions for Purchase Specifications

Mr. Yashuki Kawashima
Hokushin Electric Works, Ltd.

APPENDIX J-I

LIST OF REGISTRANTS

THE 1978 JAPANESE REGIONAL MEETING

INTERNATIONAL PURDUE WORKSHOP ON INDUSTRIAL COMPUTER SYSTEMS

Japan Electronic Industry Development Association (JEIDA)
Tokyo, Japan
June 29-30, 1978

Chichibu Cement Co.

Takumi Saito

Chiyoda Chemical Engineering & Construction Co., Ltd., 1580 Tsurumi-cho, Tsurumi-ku, Yokohama City, Kanagawa Perfecture

Norihiro Akiyama

Takashi Toyama

Chiyoda Jyohokiki Co., Ltd.

Kazuchika Ozaki

Computer Applications Co., Ltd.

Tsuyoshi Kikuchi

Electrotechnical Laboratory, 2-6-11 Nagata-cho, Chiyoda-ku, Tokyo

Osami Nomura

Kohei Sato

Koji Yada

Fuji Facom Corporation,

Hiroshi Adachi

Toshihiro Harada

Michihiro Takai

Akira Shirashima

Fuji Electric Co., Ltd., 1 Fujimachi, Hino City, Tokyo

Kyosuke Nakamura

Hitachi, Ltd., 882 Ichige, Katsuta City, Ibaragi Perfecture

Toshio Kinoshita

Kiyomi Mori

Isao Yasuda

Hokushin Electric Works, Ltd., 3-30-1 Shimomaruko, Ohta-ku, Tokyo

Koki Kawashima

Koichi Mori

Yuzo Tachikawa

Kazuo Mukai

Japan Business Automation Co., Ltd.

Taeko Hayashi

Hiroshi Hirose

Kyokuto Oil Industry Co.

Hiromi Araake

Toshio Kudo

Seiichi Yoshizawa

Kyowa Hakko Kogyo Co.

Tetsuro Sakamoto

Noboru Tohsaka

Meidensha Electric Mfg. Co., Ltd., 2-1-2 Ohtemachi, Chiyoda-ku, Tokyo

Tadamasa Baba

Tatsumi Deido

Koichi Miyashita

Masao Osawa

Eiji Sakurai

Osamu Tsuji

Mitsubishi Electric Corp., 325 Kamimachiya, Kamakura City, Kanagawa Prefecture

Kenji Hashimoto

Tadashi Hatano

Tatsuya Muto

Katsuaki Yonezawa

Mitsubishi Heavy Industry Co., 5-1 Marunouchi, Chiyoda-Ku, Tokyo

Keiichi Kobayashi

Nippon Denso Co., Ltd.

Kenzo Ito

Nippon Electric Co., Ltd. 4-14-2 Shiba, Minato-ku, Tokyo

Kazuo Manabe

Shiji Takahashi

Nippon Mini-Computer Corp.

Katsuro Sugimoto

Oki Electric Industry Co., Ltd., 4-10-3 Shibaura, Minato-ku, Tokyo

Tamio Namioka

Takeshi Ohtsuka

<u>OMRON Tateishi Electronics Co.</u>

Koji Kobayashi

<u>Pana-Facom Ltd</u>., 2-10-6 Jiyugaoka, Meguro-ku, Tokyo

Ichiyo Shirai

<u>Police of Kanagawa Prefecture</u>

Kunio Kasai

<u>Shinko Electric Co., Ltd.</u>

Yoshiaki Hikami

Hidemitsu Tabata

<u>Sky Aluminum Co., Ltd</u>.

Takao Suenaga

<u>Software Development Co</u>.

Tetsuo Osaki

<u>Sumitomo Chemical Co</u>., 1-2-2 Shinsenri-Nishimachi, Toyonaka City, Aichi Prefecture

Hiroshi Nishikawa

<u>Sumitomo Heavy Industries Co., Ltd</u>.

Munetaka Jyotaki

<u>Toa Nenryo Kogyo Co</u>.

Hideaki Sugawara

<u>Tokyo Gas Co</u>.

Kunihiro Mori

<u>Tokyo University</u>, 7-3-1 Hongo, Bunkyo-ku, Tokyo

Yasushi Kida

<u>Tonen Sekiyu Kagaku K.K.</u>

Katsuyoshi Hosoya

<u>Toshiba Corporation</u>

Tetsunori Kai

Yoshihiro Matsumoto

Katsuji Shimokawa

Akihiro Uetani

Wasei Yamanaka

<u>Toshiba Engineering Co</u>.

Kohei Yonetani

<u>Toshiba Machine Co</u>.

Akira Watanabe

<u>Waseda University</u>

Seinosuke Narita

<u>Yamatake Honeywell Co., Ltd</u>., 1-12-1 Kawana, Fujisawa City, Kanagawa Prefecture

Toshio Nakagawa

Hikaru Ogawa

Shinichi Shigehiro

Masao Sugita

Masaaki Tojyo

Masaya Usui

<u>Yasukawa Electric Mfg. Co., Ltd</u>.

Hirotoshi Sese

<u>Yokogawa Electric Works, Ltd</u>. 2-9-32 Naka-cho, Musashino City, Tokyo

Yoshiji Fukui

Yoshihide Murakakami

Yoshio Tomita

Shigeru Yamamoto

Michio Yoshioka

SECTION III

MINUTES OF THE AMERICAN REGIONAL MEETING INTERNATIONAL PURDUE WORKSHOP ON INDUSTRIAL COMPUTER SYSTEMS

MINUTES OF THE AMERICAN REGIONAL MEETING

INTERNATIONAL PURDUE WORKSHOP ON INDUSTRIAL COMPUTER SYSTEMS

Purdue University

April 10-12, 1978

1. <u>ORGANIZATION</u>

The 1978 Spring Regional Meeting of the American Region of the International Purdue Workshop on Industrial Computer Systems took place in Room 310, Stewart Center, Purdue University, West Lafayette, Indiana, on April 10-12, 1978. The Agenda of Insert A-I was followed and those individuals listed in Appendix A-I were registered for the meeting.

2. <u>TUTORIAL</u>

As listed in the Agenda, four separate major tutorial lectures were presented at this Workshop. They were:

1. Mr. Thomas C. Pingle
   Western Electric Company

   "SPIDER - A Hierarchical Industrial Manufacturing and Control System"

2. Mr. Charles W. Bachman
   Honeywell Information Systems

   "Distributed Systems Reference Model"

3. Mr. Remsi Messare
   Exxon Research and Engineering Company

   "Tentative Functional Requirements and Design Criteria for Operator Interfaces"

4. Lt. Col. William A. Whitaker
   Chairman - DOD-HOL Working Group

   "Update on IRONMAN"

All were very much appreciated by the Workshop attendees. We wish to thank each of the speakers most sincerely for his presentation.

3. RESOLUTIONS

Because of the continuing discussion between the FORTRAN Committees of Purdue Europe and Purdue America, the Workshop meeting passed the two resolutions of Inserts A-II and A-III. This action is intended to help resolve any conflicts which might arise between Committees as well as between Regional Branches of a single Committee. An example would be to have equivalent tasking requirements for FORTRAN, BASIC and the LTPL as far as the language definitions permit.

Insert A-IV, printed in green, presents a proposal of the TC5-A Committee on Interfaces and Data Transmission concerning the frame structure for the data link control structures to be used in communication links between devices in a distributed process control system.

INSERT A-I

AGENDA

FIFTH AMERICAN REGIONAL MEETING

INTERNATIONAL PURDUE WORKSHOP

ON INDUSTRIAL COMPUTER SYSTEMS

Room 310
Stewart Center
Purdue University
West Lafayette, Indiana 47907

April 10-12, 1978

<u>Monday</u>, <u>April</u> <u>10</u>, <u>1978</u>

AM

08:30 - 09:00 Registration - Room 310, Stewart Center

09:00 - 09:05 Introduction and Discussion of Workshop Program

09:05 - 09:30 Status of Standardization Efforts of this Workshop and Others on Language and Hardware Standardization for Industrial Computer Systems.

Mr. Richard H. Caro, Chairman,
American Region, for
Dr. Thomas J. Harrison,
Workshop Vice Chairman for Standards

09:30 - 09:45 Report on Results of the European Regional Meeting

Dr. Theodore J. Williams
Workshop Chairman, and
Mr. Richard H. Caro

09:45 - 10:00 Coffee Break

10:00 - 11:30 "SPIDER - A Hierarchical Industrial Manufacturing and Control System"

Mr. Thomas C. Pingle
Department Chief - Computer Aided Manufacture
Western Electric Company
4513 Western Avenue
Lisle, Illinois 60532

INSERT A-I (Cont.)

PM	
11:30 - 12:30	Presentations by Committee Chairpersons of the Mission, Accomplishments, and Planned Programs of the Several Workshop Committees. Industrial Real-Time FORTRAN Committee Dr. Matthew R. Gordon-Clark Chairman Long Term Procedural Languages (American) Committee Mr. Merritt E. Adams, Chairman Interface and Data Transmission Committee Dr. R. Warren Gellie, Chairman Man/Machine Interface Communications Committee Mr. Robert F. Carroll, Chairman Systems Reliability, Safety, and Security Committee Mr. Steve Hussar, Acting Chairman Microprocessor Ad Hoc Committee Mr. Yoel Keiles
12:30 - -1:30	Lunch
01:30 - 02:30	"Distributed Systems Reference Model" Charles W. Bachman Honeywell Information Systems Deer Valley Park Box 6000, Station C61 13430 Black Canyon Highway Phoenix, Arizona 85005
02:30 - 02:45	Coffee Break
02:45 - 05:00	Organization of Workshop Committees and Scheduling of Committee Meetings.

INSERT A-I (Cont.)

05:00 - Close Meeting of the USTAG, ISO/TC97/SC5/WG1

Dr. Matthew Gordon-Clark, Chairman

Tuesday, April 11, 1978

AM
09:00 - 10:00 "Tentative Functional Requirements and Design Criteria for Operator Interfaces"

Mr. Remsi Messare
Senior Project Engineer
Exxon Research and Engineering Company
Box 101
Florham Park, N.J.

10:00 - 10:15 Coffee Break

10:15 - 11:00 "Update on Ironman"

Lt. Col. William A. Whitaker
Chairman - DOD-HOL Working Group
Defense Advanced Research Projects Agency
1400 Wilson Blvd.
Arlington, VA 22209

PM
11:00 - Close Meetings of Workshop Committees as Scheduled by Committee Chairpersons.

Wednesday, April 12, 1978

AM PM
08:00 - 01:00 Meetings of Workshop Committees as Scheduled by Committee Chairpersons.

01:00 - 01:15 Presentation of Results of Work of the Interfaces and Data Transmission Committee

01:15 - 01:30 Presentation of Results of the Work of the Micro Processor Ad Hoc Committee

01:30 - 01:45 Presentation of Results of Work of the Man/Machine Interface Committee

01:45 - 02:00 Presentation of Results of Work of the Industrial Real-Time FORTRAN Committee

INSERT A-I (Cont.)

02:00 - 02:15	Presentation of Results of Work of LTPL-A Committee
02:15 - 02:30	Election of Officers, New Business, Discussion of Time and Length of Next Meeting
02:30 -	Close of Meeting

INSERT A-II

RESOLUTION

The Vice-Chairman of Standards of the IPW is requested to coordinate the actions of all of the technical committees of the workshop to ensure that they produce technically equivalent solutions for common problems.

INSERT A-III

RESOLUTION

The Chairman of the American Region shall be responsible for coordinating the actions of the American Region Software Committees (TC's 1, 2, 3, 4 & 8) to ensure that they produce technically equivalent solutions for common problems. To this end, the chairman, American Region, may appoint a coordinator for software standards if so required.

INSERT A-IV

FRAME STRUCTURE FOR DATA LINK CONTROL PROCEDURES

The American Region of the International Purdue Workshop on Industrial Computer Systems (IPW-A) through its committee TC5-A "Interfaces and Data Transmission" recommends the following frame structure for the data link control procedures to be used in communication links between devices in a distributed process control system.

1. Introduction

This recommendation is the first in a series of recommendations to be promulgated by IPW-A as part of its objective to generate a standard for inter-subsystem communication in distributed industrial process control systems.

2. Frame Structure

All transmissions are in frames and each frame conforms to the following structure:

Flag	Address	Control	Information	FCS	Flag
01111110	8 bits	8 bits	*	16 bits	01111110

*An unspecified number of bits but which is an integral number of octets (bytes).

Frames containing only data link control sequences form a special case where there is no Information field.

where

Flag	=	flag sequence (F)
Address	=	address field (A)
Control	=	control field (C)
Information	=	information field (I)
FCS	=	frame checking sequence

3. Elements of the Frame

3.1 Flag Sequence (F)

All frames start and end with the flag sequence. This sequence is a zero bit followed by 6 one bits followed by a zero bit (01111110). All stations attached to the data link continuously hunt, on a bit-by-bit basis, for this sequence. A transmitter

must send only complete eight-bit flag sequences, however, the sequence of 011111101111110 at the receiver is two flag sequences. The flag is used for frame synchronization.

In order to achieve transparency the flag sequence is prohibited from occurring in the Address, Control, Information and FCS fields via a "zero bit insertion" procedure described in Section 3.6.

The flag sequence which closes a frame may also be the opening flag sequence for the next frame. Any number of complete flags may be used between frames.

3.2 Address Field (A)

The address field in all cases contains station address information. The length of this field (A) is N octets (N greater than or equal to 1).

The address range can be extended by reserving the first transmitted bit of each address octet which is then set to binary zero to indicate that the following octet is an extension of the basic address field.

When extensions are used the presence of a binary "1" in the first transmitted bit of the basic address octet signals that only one address octet is being used.

3.3 Control Field (C)

The control field contains a command or response and may contain other information.

The control field may be extended by one or more octets. The extension methods and the bit patterns for the commands and responses are not defined in this recommendation.

3.4 Information Field (I)

The information field may be any sequence of bits but will be an integral number of octets.

INSERT A-IV (Cont.)

3.5 Frame Check Sequence (FCS)

All frames include a 16-bit frame check sequence (FCS) just prior to the closing flag for error detection purposes. The contents of the address, control and information fields, excluding the zeros inserted to maintain transparency as per Section 3.6, are included in the calculation of the FCS sequence. The FCS is that defined in IS 3309 for HDLC.

3.6 Transparency

Transparency is provided for data coded in the Information field. The occurrence of the flag sequence within a frame is prevented via a "Zero bit insertion" procedure as follows:

> The transmitter inserts a zero bit following five contiguous one bits anywhere between the opening flag and the closing flag of the frame. The insertion of the zero bit thus applies to the contents of the Address, Control, Info and PCS fields (including the last 5 bits of the FCS). The receiver continuously monitors the received bit stream; upon receiving a zero bit followed by contiguous one bits, the receiver inspects the following bit: If a zero, the 5 one bits are passed as data and the zero bit deleted. If the sixth bit is a one, the receiver inspects the seventh bit; if it is zero, a flag sequence has been received; if it is a one an invalid frame has been received.

3.7 Invalid Frame

An invalid frame is defined as one that is not properly bounded by two flags or one which is too short (for example, shorter than 32 bits between flags). Invalid frames shall be ignored. Thus a frame which ends with an all "1" bit sequence equal to or greater than seven bits in duration shall be ignored.

4. ELECTIONS

A Nominating Committee composed of Messrs. Robert F. Carroll and R. Warren Gellie was appointed to serve at this Workshop Meeting. They nominated Mr. Richard Caro for a second term as American Regional Chairman. He was elected unanimously.

5. COMMITTEE REPORTS

Appendices A-II to A-VI present the Reports of Committees, TC-1A, FORTRAN; TC-3A, LTPL; TC-5A, Interfaces and Data Transmission; TC-6A, Man/Machine Interfaces; and, TC-7A, Reliability, Safety and Security. There were no reports from Committees TC-2A, BASIC; TC-4A, POLS; TC-8A, Operating Systems and TC-9A, Glossary.

Mr. Yoel Keiles, Chairman of the Ad Hoc Committee on Microprocessors gave a verbal report. His committee will have a proposal for the International Workshop Meeting in October.

6. USA TAG ISO/TC97/SC5/WG1

Appendix A-VII presents the report of the USA TAG Meeting held the evening of Monday, April 10, 1978 at Purdue University as an adjunct to the Workshop meeting.

7. FUTURE MEETINGS

This Meeting was an experiment to try a three day meeting. As a result the attendees agreed to return to the previous four day schedule. Therefore the next (1979)

American Spring Regional Meeting of the Workshop will be held on April 23-26, 1979 in Room 310, Stewart Center, Purdue University, West Lafayette, Indiana.

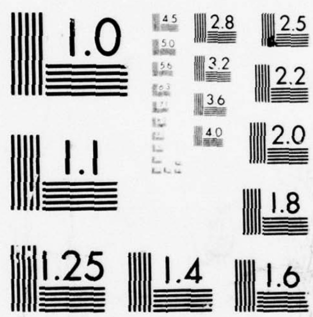
The Sixth International Meeting of the Workshop will be held on October 9-12, 1978, also at Purdue University, West Lafayette, Indiana.

AD-A068 266 PURDUE UNIV LAFAYETTE IND PURDUE LAB FOR APPLIED IND--ETC F/G 9/2
MINUTES 1978 SPRING REGIONAL MEETING INTERNATIONAL PURDUE WORKS--ETC(U)
JUN 78 N00014-78-C-0127

UNCLASSIFIED NL

2 OF 4

AD
A068266


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

APPENDIX A-I

LIST OF REGISTRANTS

FIFTH AMERICAN REGIONAL MEETING

INTERNATIONAL PURDUE WORKSHOP

ON INDUSTRIAL COMPUTER SYSTEMS

April 10-12, 1978

Purdue University
West Lafayette, Indiana

Aluminum Company of America (ALCOA),1501 Alcoa Bldg., Pittsburgh, Pennsylvania 15219 (412/553-3114)

James E. Owens, Coordinator - P. C. Computers

Applied Automation, 201 RB 2, Bartlesville, Oklahoma,74004 (918/661-3637)

Dick Thompson, Engineer

Applied Technology, 645 Alamanor, Sunnyvale, California 94086 (408/732-2710)

Richard Nielson, Programmer

Armco Steel Corp., 703 Curtis, Middletown, Ohio 45043, (513/425-2209)

Jerry H. Schunk, Senior Research Engineer

Atlantic Richfield Co., 400 E. Sibley Blvd., Harvey, Illinois 60426 (312/333-3000)

Jeffrey K. Fang, Analyst

Gary L. Troyer, Senior System Analyst

Bethlehem Steel Corp., Homer Research Laboratory, Bethlehem, Pennsylvania 18016 (215/694-3541)

Stephen Gifford, Engineer

Kenneth E. Platt, Assistant Manager, Eighth Avenue Building, Bethelehem, Pennsylvania 18016 (215/694-7442)

APPENDIX A-I (Cont.)

B. F. Goodrich Chemical, 6100 Oak Tree Blvd., Cleveland, Ohio 44131 (216/524-0200)

Robert F. Carroll, Manager Systems Engineer

Bristol Systems/Instruments, 40 Bristol Street, Waterbury, Connecticut 06712 (203/756-4451)

Walter A. Duncan, Manager Software Dev.

Cincinnati Milacron, 4701 Marburg Avenue, Cincinnati, Ohio 45209 (513/841-8719)

Richard Read

Computer Sciences Corp., 2251 San Diego Avenue, San Diego, California 92110 (714/299-9902)

(Dr.) Donald Devorkin, Senior Scientist

Corning Glass Works, Bldg. Rt., Corning, New York 14830 (607/974-7314)

Arthur Q. Thompson, Supervisor Digital System Engineering

Cutler-Hammer, Inc., 4201 N. 27 Street, Milwaukee, Wisconsin 53216 (414/442-7800)

Donald L. Ness, Senior Engineer

Defense Advanced Research Projects Agency, 1400 Wilson Blvd., Arlington, Virginia 22209 (202/694-1139)

Lt. Col. William A. Whitaker

Department of Industry Trade & Commerce, 235 Spark Street, Ottawa, Ontario, Canada K1A 0H5 (613/593-7861)

J. Scrimgeour, Consultant, Technology Branch (61)

E. I. duPont deNemours & Co., Louviers Building, Wilmington, Delaware 19898 (302/772-4296)

Robert S. Crowder, Jr., Vice Chairman - Planning Engineering Dept., duPont Exp. Sta 357

Stephen C. Schwarm, Development Engineer, (302/774-1669)

APPENDIX A-I (Cont.)

Exxon Research, Box 101 Florham Park, New Jersey 07932 (201/474-6869)

Ara Barsamian, Staff Engineer

R. R. Messare, Staff Engineer (201/474-1374)

Fischer & Porter Co., County Line Road, Warminister, Pennsylvania 18974 (215/674-8492)

Shoji Fukumoto Manager Software Development

Edward J. Rathje, Senior System Analyst (215/674-6483)

Richard P. Sanders, Manager (215/674-6493)

Forney Engineering, 3405 Wiley Post Road, Addison, Texas 75001 (214/233-1871)

Andy Evalds, Project Engineer

The Foxboro Company, 38 Neponset Avenue, Foxboro, Massachusetts 02035 (617/543-8750)

Dick Caro, Principal Research Engineer (Dept. 330)

Robert F. Sapita, Marketing Specialist (x-2233)

H. J. Shepherd, Section Manager (Dept. 127)

General Electric Co., Bldg. 25 EE, 1285 Boston Avenue Bridgeport, Connecticut 06602 (203/334-1012)

Richard W. Signor, System Software Engineer (x-1610)

General Motors, Manufacturing Development, GM Tech Center Warren, Michigan 48090 (313/575-0923)

Bob Campbell, Sr., Prof. Engineering

Gould-Modicon, Box 83, Andover, Massachusetts 01810 (617/475-4700)

Bruce Allen, Manager Futures

Tom McLeod, Product Manager

Herbster Scientific, 3233 Mangum Road #100, Houston, Texas 77092 (713/681-7757)

John F. Herbster

APPENDIX A-I (Cont.)

Hershey Foods Corp., 19 E. Chocolate Avenue, Hershey, Pennsylvania 17033 (717/273-8668)

James P. Foster, Senior Systems Control Engineer

Hewlett-Packard, 11000 Wolfe Road, Cupertino, California 95014 (408/257-7000)

Bruce Stowell, Development Engineer

Honeywell Information Systems, 300 Concord Road, Billerica, Massachusetts 02173 (617/667-3111)

Charles W. Bachman, Senior Research Scientist (x-2256)

Honeywell, Process Control Division, 1100 Virginia Drive, Fort Washington, Pennsylvania 19034 (215/643-1300)

Charles Diefenderfer, Principal Development Engineer

Stephen P. Higgins, Jr., Principal Control Engineer, 2222 W. Peoria Avenue, Phoenix Arizona 85029 (602/943-2341)

Yoel Keiles, Principal Engineer

IBM Corporation, 555 Bailey Avenue, San Jose, California 95150 (408/463-4084)

Alex J. Arthur, Staff Programmer

I/C Engineering, 3221 S. La Cienega Blvd., Los Angeles, California 90016

L. C. Boxhorn

Industrial Nucleonics Corp., 650 Ackerman Road., Columbus, Ohio 43202 (614/261-2000)

Robert G. Wilhelm, Jr., Principal Engineer

Inland Steel Co., 3210 Watling Street, East Chicago, Indiana 46312 (219/392-5672)

Mark S. Borowiak, Engineer

Randall Pals, Engineer (219/392-4066)

Intel, 3065 Bowers Avenue, Santa Clara, California 95051 (408/987-7432)

John Zarrella, Senior Engineer

APPENDIX A-I (Cont.)

International Harvestor, 401 N. Michigan, Chicago, Illinois 60611 (312/836-2183)

L. D. Gilmore, Manager Engineer & Process Control

Jet Propulsion Laboratory, 4800 Oak Grove Drive, Pasadena, California 91103 (213/354-6127)

Fred T. Krogh

William V. Snyder

Lawrence Berkeley Laboratory, Berkeley, California 94720 (415/843-2740)

R. F. Thomas, Comp. Scientist

Leeds & Northrup, Dickerson Road, Northwales, Pennsylvania 19454 (215/643-2000)

Richard Flath, Application Consultant

Simon Korowitz, Sr. Project Engineer (x-8518)

Measurex Corporation, 1 Results Way, Cupertino, California 95014 (408/255-1500)

John D. Higham, Manager Software & Control

Roger M. Lee, Staff Engineer

Metromation Inc., 1101 State Road, Princeton, New Jersey 08648 (609/924-3900)

R. A. Williamson, Jr., Marketing Manager

Middle South Services, Inc., 225 Baronne Street, New Orleans, Louisiana 70112 (504/529-5262)

John D. Stenquist, Senior Analyst

Modcomp, 1650 W. McNabro, Fort Lauderdale, Florida 33309 (305/974-1380)

Samuel L. Miles, Real Time Product Manager

Monsanto, 800 N. Lindberg, Saint Louis, Missouri 63011 (314/694-6506)

V. A. Lauher, Fellow

APPENDIX A-I (Cont.)

National Air Development Center, Warminster, Pennsylvania 18974

W. Norr

National Research Council/Canada, Control Systems Lab, Ottawa Canada KIA OR6 (613/993-2808)

(Dr.) Warren R. Gellie, Research Officer

Naval Ocean Systems Center, Code 8221, 6542 Alcala Knolls Drive, San Diego, California 92111 (714/225-2480) (sec.-7282)

Warren E. Loper, Computer Specialist

Naval Weapons Center, China Lake, California 93555 (714/939-3564)

R. D. Hawkins, Microprocessor Design Coordinator

Olin Corporation, 120 Long Ridge Road., Stanford, Connecticut 06904 (203/356-2719)

Alex Habib, Automation System Manager

Owens Corning Fiberglass, P. O. Box 415, Granville, Ohio 43023 (614/587-0610)

Mr. Bissell, Senior Research Associate (x-700)

PPG Industries, Inc., One Gateway Center, 9 West, Pittsburgh, Pennsylvania 15222 (412/434-3341)

Stephen G. Hussar, Senior Process Control Engineer

Purdue University, West Lafayette, Indiana 47907 (317/494-8425)

T. J. Williams, Professor of Engineering and Director Purdue Laboratory for Applied Industrial Control

Scott Paper Company, Scott Plaza, Philadelphia, Pennsylvania 19113 (215/521-5000)

Matthew Gordon-Clark, Scientific Associate (x-2359)

David L. Legrow, Senior Research Project Engineer

The Standard Oil Company (Ohio), Midland Bldg., Cleveland, Ohio 44115 (216/575-5516)

Donald G. Kempfer, Sr. Tech. Sp.

APPENDIX A-I (Cont.)

Tektronix, Box 500, M/S 50/454, Beaverton, Oklahoma 97077
(503/644-0161)

Maris Graube (x-6234)

Texas Instruments, P. O. Box 2909, M/S 2203, Austin, Texas
78769 (512/258-7925)

Carl Wilson, Section Manager

U. S. Steel Corporation, Research Laboratory, 125 Jamison Lane,
Monroeville, Pennsylvania 15146 (412/372-1212)

David H. Reed, Senior Research Engineer

Waterloo, University of, Waterloo, Ontario, Canada N2L 3G1
(705/885-1211)

Warren Little, Associate Professor

Western Electric, 4513 Western Avenue, Lisle, Illinois 60532
(312/393-5149)

M. E. Adams, SISSM

Thomas C. Pingel, Department Chief (312/983-3536)

Westinghouse Electric Corporation, 1441 W. Alameda Drive,
Tempe, Arizona 85282 (602/968-3171)

Michael R. Doyle, Headquarter Sales (x-218)

APPENDIX A-II

TC-1

INDUSTRIAL REAL-TIME FORTRAN COMMITTEE

APPENDIX A-II

International Purdue Workshop
IPW
Industrial Computer Systems

PURDUE LABORATORY FOR
APPLIED INDUSTRIAL CONTROL
102 Michael Golden
Purdue University
West Lafayette, Indiana 47907, USA
317/494-8425

Please reply to:

April 20, 1978
Scott Paper Company
Scott Plaza III
Philadelphia, PA 19113

Minutes of the Joint Meeting of ISA SP61 committee and TC-1A committee of International Purdue Workshop held at Purdue University, Lafayette, Indiana, April 10-12.

Those present

Matthew R. Gordon-Clark	Scott Paper Company
Mark S. Borowiak	Inland Steel Company
Richard H. Caro	Foxboro
Stephen G. Hussar	PPG Industries
Richard W. Signor	General Electric Company
Robert G. Wilhelm, Jr.	Industrial Nucleonics Corp.

It was agreed that the initial meeting of the committee would be held jointly with TC-3A. The topics discussed in the joint meeting were the report of Dick Caro on the recent meeting of IPW Europe and the recent meetings of ANSI X3J3 (Rich Signor) and of ANSI X3J1 (Alex Arthur). The minutes of the joint meeting will be issued by Alex Arthur secretary of TC-3A.

The committee devoted its entire time to the consideration of dp ISA S61-3 March 1978. Dick Caro had spent considerable time at Purdue Europe discussing the differences between IPWE Real Time FORTRAN and S61-3. The primary differences concern events, eventmarks, semaphores, interrupts. At the meeting of TC-E in Zurich, it had been found that the eventmark concept agreed upon at the October meeting of the IPW which was considered to cover both process interrupts and semaphores was unsatisfactory (Eventmarks as defined in dp ISA S61-3 March 1978).

The central problem is one of function. For interrupts, when the event occurs, it is required that all tasks waiting for the event should be activated and the operating system will handle by priority or other mechanisms the exact order of execution. For example, if a pump fails, the tasks initiated could include, start a back-up pump, print a log of the last five minutes, and display suitable alarm.

For semaphores, when the event occurs, it is required that only one waiting task be activated and the operating system must decide by some mechanism such as priority, FIFO queue etc. which task will be activated. For example if a semaphore is used to control a resource such as a printer only one task can use the printer at any time.

Affiliations
Purdue University
Instrument Society of America through Data Handling and Computations, Chemical and Petroleum Industries, and Automatic Control Divisions
International Federation for Information Processing as Working Group WG5-4. Common and/or Standardized Hardware and Software Techniques of Technical Committee, TC-5, Computer Applications in Technology
Institute of Electrical and Electronic Engineering, Data Acquisition and Control Committee of the Computer Society, and Industrial Control Committee of the Industrial Application Society
International Federation of Automatic Control, Computer Committee
National Research Council of Canada, Associate Committee of Automatic Control
Commission of the European Communities (CEC) through its Directorate-General for Industrial and Technological Affairs
Japan Electronic Industry Development Association (JEIDA) through its IPW Japan Committee

APPENDIX A-II (Cont.)

Minutes of Meeting
April 20, 1978
Page 2

The following changes were made to dp ISA S61-3 March 1978:

a) Eventmarks (Section 3) will have three states only. Undefined, ON, OFF.

b) Eventmarks will activate all tasks waiting on the event.

c) Two subroutines for eventmark handling will be SETEM and CLREM.

d) Initially eventmarks will be undefined.

e) A new section (Section 4) will cover resource marks.

f) Resource marks will activate one and only one task waiting on the event.

g) Two subroutines for resource mark handling will be LOCK and UNLOCK.

h) Present Section 4 will become Section 5.

A draft of Section 4 (Resource marks) and a rewrite on Section 3 (Eventmarks) was agreed upon.

The document dp ISA S61-3 March 1978 was reviewed for errors and other corrections; notably the references to FORTRAN will refer to ANS X3·9-1978; the references to Hollerith in Appendix B will be changed; and the program name definitions will be brought in line with the new standard FORTRAN.

Actions

Rewrite Section 3 Eventmarks	Matthew Gordon-Clark
Write Section 4 Resource marks	Rick Signor
Define new name parameter	Matthew Gordon-Clark
Appendix on use of name parameter	Rick Signor
Typographical and other errors	All

All these actions to be completed and sent to Dick Caro at Foxboro by 25 April 1978. An updated version of dp ISA S61-3 will be available by May 15, 1978.

Meetings

Philadelphia	June 5/6, 1978
Purdue	October 9-12, 1978

San Francisco area in conjunction with TC1-A in January or February 1979.

Matthew R. Gordon-Clark

Matthew R. Gordon-Clark
Chairman of TC1-A
Chairman of ISA SP61

APPENDIX A-III

TC-3

LONG TERM PROCEDURAL LANGUAGES COMMITTEE

APPENDIX A-III

LTPL-A/59
78/04/28

To: LTPL-A Members

78/04/28
Alex J Arthur
IBM Corp, M77/G20
555 Bailey Ave.
San Jose, CA 95150, USA

LTPL-A Meeting at International Purdue Workshop Americas Regional Meeting, 78/04/10-12

The FORTRAN and LTPL-A committees decided to initially hold a joint meeting. The attendance list is enclosed. The agenda agreed for this meeting was

Organisation
Tasking
S61.3
DoD Review

ORGANIZATION

The remaining FORTRAN work and the major part of the LTPL work concern tasking. TC-2 (BASIC) are also into the same problem area. Positive co-ordination is therefore required. Various alternative approaches were considered. The concensus was however to do nothing formal at the committee level but to leave it to the Vice Chairpersons for Standards and the Americas Region to do the appropriate things.

TASKING

There are 3 separate state diagrams around for tasking.

1) Operating Systems committee, p. 297 of minutes of International Meeting, 1977.

2) Industrial Real Time FORTRAN.

3) S61.3

PRECEDING PAGE BLANK - NOT FILMED

(1) has INC and DEC. They operate on a <u>counter</u> whose range is (-:0:+).

(2) IRT FORTRAN has a number of functions operating on a <u>semaphore</u> whose range is (0:+). SIGNAL adds to it, WAIT decrements it. They have WAIT on EVENT, WAIT on time, AWAIT (hardware interrupt).

(3) S61.3 has <u>event marks</u>. They are counters whose range is (0:+). There is an incrementing function, SETEM, a zeroing function, CLREM, a decrementing function, WAITE on event mark, and a WAIT on time function.

In IRT FORTRAN a semaphore is defined such that <u>only</u> one waiting program runs when a semaphore is SIGNALED.

If S61.3 is modified to allow an event mark to go negative then SETEM and WAITE correspond to INC and DEC and semaphores and monitors can then be built up just as in the Operating Systems paper. Caro then proposed to modify S61.3 by changing SETEM and CLREM to INCEM and DECEM, allowing event marks to be negative, and adding a POST function.

DeVorkin was concerned that there seemed to be 2 different types of task queueing involved, do all and do fifo and that he was not sure how to do both with the mechanism proposed. After some further discussion he seemed to be satisfied that for example the mechanism can be reduced to simply POST and WAITE for the do all case. Matthew Gordon-Clark felt quite strongly that users are more concerned with the ability to start or recommence tasks on interrupts rather than concern themselves with task synchronisation.

A table was built to contrast the use of EVENT as an event mark rather than as a sync signal.

1. CALL POST (e,m)	CALL SIGNAL (s,m)
2. CALL WAITE (e,m)	CALL SYNC (s,m)
3. CALL CLREM (e,m)	

It was agreed that UNLOCK and LOCK seemed better words than SIGNAL and SYNC. Also it did not seem reasonable to handle all the possible complications of ABORT in S61.3 and that there should simply be a paragraph stating clearly that the problems had not been addressed in the document.

S61.3

The remaining problem between S61.3 and the IRT FORTRAN proposals is that IRT FORTRAN inherits pre-existing state and S61.3 does not. At this point, it seems as though the groups have agreed to disagree.

Dick Caro then brought to the meeting's attention some differences between TC-1E and TC-1A relative to ISA S61.2-1978.

TC-1E's main concern had been access privilege with the assumption that successful completion of a call had changed the state of the file whereas TC-1A had been interested mainly in access mode by the calling program. A conflict matrix was then constructed.

calling task							
		wants to					
will allow		READ	Yes	Yes	No	No	File Is
others to		WRITE	Yes	No	Yes	No	
READ	WRITE						
Yes	Yes		Unlock	Unlock	Unlock	X	Unlocked
Yes	No		Update (1)	Prot Read	Update (2)	X	Protected
No	Yes		X	Unlock	X	X	Unprotected
No	No		Lock	Lock	Lock	X	Locked

If we change PROTECTED READ to PROTECTED then case (1) and (2) are both covered.

REPORTS

Rick Signor reported on the progress of X3J3, the ANSI FORTRAN standards committee. The new FORTRAN standard is now official. The committee has started work on the 1982 revision. To this end they are

having a series of presentations on possible extensions and are establishing liason with other interested groups. These include CODASYL, IPW, numerical analysts, vector and array processing, D D and ERDA Standards.

The committee is establishing the scope of FORTRAN 82. They are trying to identify problems and new requirements such as the need for data structures, data base, industrial control, tasking, basic data types, subroutine interface improvements, extension methodology (define core language + extensions), guidelines for change to language. Keywords in CALLs looks like the right thing to propose.

Alex Arthur gave a short report on the progress of the ANSI subcommittee on real time extensions to PL/I. They now have a working document covering the full scope of the extension they will consider and are working on change and refinement of that document.

The FORTRAN committee went off to do the work of revising S61.3 to incorporate the agreed upon changes.

DoD REVIEW

The LTPL-A committee decided that it was premature to further consider the DoD languages, considering that the DoD was about to select 2 of the languages to go into the next phase and we would either have to guess which two or would do unnecessary work.

Also the DoD expect the two successful contractors to fill out their language specifications based on the comments by the 80 reviewing bodies and report back before starting to implement their prototype compilers. At that point, the revised specifications will be made public and it seems appropriate then for LTPL-A to meet and review them relative to our earlier comments. Also at that time STEELMAN and PEBBLEMAN will be available for review. An LTPL-A meeting will therefore be held in San Diego, July 17-19, 1978.

Alex J. Arthur
Secretary, LTPL-A

APPENDIX A-III (Cont.)

LTPL-A/59 PAGE 5

ATTENDEES

NAME	AFFILIATION
M. E. Adams	Western Electric
A. J. Arthur	IBM Corp.
M. S. Borowiak	Inland Steel
R. H. Caro	Foxboro
D. Devorkin	Computer Sciences Corp.
W. A. Duncan	Bristol Systems
R. Flath	Leeds and Northrup
M. R. Gordon-Clark	Scott Paper
R. D. Hawkins	Naval Weapons Center
J. D. Higham	Measurex
S. G. Hussar	PPG Industries
F. T. Krogh	JPL
W. Little	Univ. of Waterloo
W. E. Loper	Naval Ocean Systems Center
R. Nielson	Applied Technology
E. J. Rathje	Fischer and Porter
S. C. Schwarm	Dupont
J. Scrimgeour	Canadian DITC
R. W. Signor	General Electric
W. V. Snyder	JPL
J. D. Stenquist	Middle South Services, Inc.
B. Stowell	Hewlett Packard
R. F. Thomas	Lawrence Berkeley Laboratory
G. L. Troyer	Atlantic Richfield
W. A. Whitaker	DARPA
R. G. Wilhelm	Industrial Nucleonics
C. Wilson	Texas Instruments

APPENDIX A-IV

TC-5

INTERFACES AND DATA TRANSMISSION COMMITTEE

International Purdue Workshop
IPW
Industrial Computer Systems

PURDUE LABORATORY FOR
APPLIED INDUSTRIAL CONTROL
102 Michael Golden
Purdue University
West Lafayette, Indiana 47907, USA
317/494-8425

Please reply to:

TC5-A: Interfaces and Data Transmission Committee

Committee Meeting held at Purdue during Spring Regional Workshop

April 10-12, 1978

Summary of Activities

Attendance

Twenty-nine members were present. A list of names and organizations is attached.

TC5-A and SP72

Some time was spent in discussion of the dual role of TC5-A and SP72. It was agreed that the committee could contribute most effectively in the development of a standard industrial communication subsystem by confining its SP72 activities to the TAG responsibility to the IEC SC65A/WG6 experts, and spending most of its effort as TC5-A in developing recommendations and guidelines leading toward a standard.

IEC SC65A/WG6

Bob Crowder introduced the revised WG6 functional requirements document "PROWAY". He explained that the intent of the revision was to improve the editorial quality, to employ standard terms and definitions, to make the document more usable as a format for evaluating systems, and to make the requirements more specific.

It was agreed that, although this new document is significantly different in format and presentation, it describes essentially the same system as that identified by the previous TC5-A functional requirements. After a detailed examination the committee recommended no substantive changes.

Responses from Members

A number of contributions were received from members in response to the request at the last meeting for views on (i) the previous mission responses, (ii) the suitability of an HDLC-based protocol for industrial applications, and

Affiliations
- Purdue University
Instrument Society of America through Data Handling and Computations, Chemical and Petroleum Industries, and Automatic Control Divisions
International Federation for Information Processing as Working Group WG5-4. Common and/or Standardized Hardware and Software Techniques of Technical Committee, TC-5, Computer Applications in Technology
Institute of Electrical and Electronic Engineering, Data Acquisition and Control Committee of the Computer Society, and Industrial Control Committee of the Industrial Application Society
International Federation of Automatic Control, Computer Committee
National Research Council of Canada, Associate Committee of Automatic Control
Commission of the European Communities (CEC) through its Directorate-General for Industrial and Technological Affairs
Japan Electronic Industry Development Association (JEIDA) through its IPW Japan Committee

(iii) questions to be submitted to Dr. Funk. The responses, suitably "laundered", were circulated together with a report on them prepared by the Chairman. These documents are attached.

During discussion the following decisions were taken:

(i) Scheduled deadlines should be generated to assist in meeting the goal of a complete set of guidelines/recommendations within two years.

(ii) There is virtually unanimous agreement within TC5-A that an HDLC-based protocol should be recommended.

(iii) Harvey Shepherd will undertake to obtain clarification of the Funk work. He may be able to persuade Dr. Gallagher to perform this service for us.

(iv) Bob Crowder will arrange through WG6 to get all the Funk references.

(v) There is concern at the proliferation of UART-based asynchronous busses. ISA and other groups will be informed of this problem. See attached document.

(vi) TC5-A does not consider the formal evaluation and comparison of "Candidate" systems to be an essential activity of the committee. TC5-A will continue to examine and evaluate existing and proposed implementations in the course of its work, but will not follow the formalized approach adopted by WG6.

<u>HDLC</u>

The committee took a major step forward in its activity by issuing a recommendation that the HDLC frame structure should be the basis of the link level protocol in industrial inter-subsystem communication. In effect the recommendation (attached) specifies the minimum requirements which must be met in order to use currently available "SDLC" chips. The motion was passed as both a TC5-A and an SP72 recommendation. The TC5-A motion was later endorsed by the Workshop.

<u>Medias and Modulations</u>

Maris Graube presented an excellent tutorial in which he outlined the options to be considered and questions which must be addressed when defining the physical, mechanical, and electrical specifications of a communication system.

Maris also raised the issue of intrinsic safety for cables. Currently there are no well defined requirements used by the certification agencies. The committee recognized the problem and agreed that steps should be taken to remedy the situation. It was proposed that other bodies such as the System Reliability, Safety and Security Committee of IPW and ISA be informed of the problem.

Analysis of HDLC Protocol

A tabulation of message sequences and total bit requirements for various PROWAY operations using HDLC (half-duplex, normal response mode) was examined. Although there was disagreement with some of the sequences proposed, the tabulation was of value in providing members with a "feel" for HDLC and it also emphasized that HDLC, as presently defined, does not provide for transfer of mastership.

Bachman Presentation

Mr. Chas. Bachman, Chairman of the ANSI SPARC committee on distributed systems gave a Workshop presentation of the reference model being developed. His analysis in terms of concepts, entities and objects seemed to be a useful approach. However, it was apparent that the requirements, as seen by "EDP types", to be satisfied at the various levels of protocol do not include some essential needs for industrial applications.

Future Activities

Discussion of HDLC for use in industrial computer systems raised the following as areas for further investigation:

Error handling
Transfer of mastership, contention
Physical/electrical requirements
Throughput, turnaround
Frame specification
Demand handling
Level 3 (highway) functions.

Japanese Implementations

Two systems in Japan, TOSWAY and CENTUM, employ an HDLC frame structure. Following the recommendation to use an HDLC-based protocol the committee agreed that these systems should be examined in detail and used as a starting point for further activity. It was decided that at the next meeting a list of detailed questions would be prepared which would be submitted to TC5-J with a request that responses be received in time for the Fall Workshop. Further, it was proposed that TC5-J should be informed of our intentions and requested to send an expert to the Fall Workshop to further clarify their responses.

Sub-Committees

Four sub-committees were established as follows:

1. Japanese implementations: (Chairman Ara Barsamian). To prepare a detailed presentation on the Japanese systems for the next meeting.

2. Mastership transfer: (Chairman Harvey Shepherd). To examine the various approaches to extending HDLC to include this and other capabilities for industrial applications. Also to approach ANSI and ISO to see if these extensions can be integrated into the ADCCP and HDLC descriptions.

3. Frame Structure: (Chairman Dan Sze). To determine the most suitable structure within the message frame to support the requirements for industrial systems.

4. Physical/electrical: (Chairman Maris Graube). To develop the physical and electrical specifications of the communication system.

TC5-E Extended SIR Proposal

Warren Gellie hopes to meet with the new TC5-E chairman Ivor Hooton in June to discuss cooperation and coordination of the regional TC5 committees. Members were asked to examine the latest SIRE proposal (January mailing), and the attached additional comments from Hooton, and submit comments before the end of May.

Documents

It is proposed that all committee working papers will be put on microfiche. In this way all documents will be available for reference during meetings and members will be able to obtain information on previous committee activity.

Due to the rapid expansion in membership, future mailings will not include reference material unless it is specifically referenced in the minutes. Those attending the meetings will already have copies, other members will be able to obtain the documents on microfiche.

Next Meeting

The next meeting will be held July 10-12, 1978 at Honeywell, Fort Washington, PA.

R. W. Gellie,
Chairman

RWG:mf

Encl.

International Purdue Workshop

IPW

Industrial Computer Systems

PURDUE LABORATORY FOR
APPLIED INDUSTRIAL CONTROL
102 Michael Golden
Purdue University
West Lafayette, Indiana 47907, USA
317/494-8425

Please reply to:

TC5A: Interfaces and Data Transmission Committee

Committee Meeting, April 10-12, 1978

Attendance List

Bruce Allen	Gould-Modicon
Ara Barsamian	Exxon Research
Clint Boxhorn	I/C Engineering
Bob Campbell	General Motors, Mfg. Div.
Bob Crowder	DuPont
Chuck Diefenderfer	Honeywell PCD
Mike Doyle	Westinghouse Electric
Andy Evalds	Forney Engineering
Jeffrey Fang	Atlantic Richfield
Richard Flath	Leeds & Northrup
Warren Gellie	N.R.C. Canada
Maris Graube	Tektronix
Bob Hawkins	Naval Weapons Center
John Herbster	Herbster Scientific
Simon Korowitz	Leeds & Northrup
Roger Lee	Measurex
Dave LeGrow	Scott Paper
Tom McLeod	Gould-Modicon
Sam Miles	Modcomp
Donald Ness	Cutler-Hammer
James Owens	Alcoa
Randall Pals	Inland Steel
Tom Pingle	Western Electric
Richard Read	Cincinnati Milacron
Dick Sanders	Fischer & Porter
Jerry Schunk	Armco Steel
Jack Scrimgeour	Industry, Trade and Commerce, Canada
Harvey Shepherd	Foxboro
Art Thompson	Corning Glass

International Purdue Workshop
Industrial Computer Systems

PURDUE LABORATORY FOR
APPLIED INDUSTRIAL CONTROL
102 Michael Golden
Purdue University
West Lafayette, Indiana 47907, USA
317/494-8425

Please reply to:

National Research Council,
Division of Mechanical Engineering,
Bldg. M-3, Montreal Rd.,
Ottawa, Ont. K1A 0R6

TC5A: Interfaces and Data Transmission Committee

8 May 1978

SUBCOMMITTEES

1. Japanese Implementations:

Chairman: Mr. Ara Barsamian 201-474-6869
Exxon Research, Box 101, Florham Park, NJ 07932

Members: Bob Crowder 302-772-4296

2. Mastership:

Chairman: Mr. Harvey Shepherd 617-543-8750
The Foxboro Co., D-127, 38 Neponset St., Foxboro, MA 02035

Members:	Chuck Diefenderfer	215-643-1300 X426
	Simon Korowitz	215-643-2000 X8518
	Sam Miles	305-974-1380
	Dick Read	513-841-8719

3. Frame Structure:

Chairman: Dr. Dan Sze 305-994-3095
IBM Corp., GSD Dept. 22Z031-3, N.W. 51st Street,
Boca Raton, FLA 33432

Members:	Ara Barsamian	201-474-6869
	Bob Crowder	302-772-4296
	Dave LeGrow	215-521-5000
	Dick Read	513-841-8719
	Dick Sanders	215-674-6493

4. Physical/Electrical:

Chairman: Mr. Maris Graube 503-644-0161 X6234
Tektronix, Box 500, M/S 50/454, Beaverton, OK 97077

Members:	Bob Campbell	313-575-0923
	Don Ness	414-442-7800

Affiliations
Purdue University
Instrument Society of America through Data Handling and Computations, Chemical and Petroleum Industries, and Automatic Control Divisions
International Federation for Information Processing as Working Group WG5-4. Common and/or Standardized Hardware and Software Techniques of Technical Committee, TC-5, Computer Applications in Technology
Institute of Electrical and Electronic Engineering, Data Acquisition and Control Committee of the Computer Society, and Industrial Control Committee of the Industrial Application Society
International Federation of Automatic Control, Computer Committee
National Research Council of Canada, Associate Committee of Automatic Control
Commission of the European Communities (CEC) through its Directorate-General for Industrial and Technological Affairs
Japan Electronic Industry Development Association (JEIDA) through its IPW Japan Committee

APPENDIX A-V

TC-6

MAN/MACHINE COMMUNICATIONS COMMITTEE

APPENDIX A-V

MEETING MINUTES

April 12, 1978

Device Independent Guidelines

The tasks performed at a MMIF were reviewed to determine their completeness (see attached lists).

Each task is to be examined as to what type data must be passed (i.e. commands, control parameters and verification feedback). Several tasks were examined and the form of reporting was defined. Each member will review his share of these tasks and send in the results for tabulation. The result may determine if a device independent language is practical.

Reasons for Implementing or Not Implementing Modern Man-Machine Interface Functions

Those reasons both pro and con were reviewed. A new writeup of the positive and negative factors were derived. These were distributed to the members for comments. See attached copy.

Guideline Update

It was agreed that we were at a point at which to improve the guidelines. There are several areas that need attention:

1. The bibliography section needs to be classified and categorized so that it is more usable for the reader (enclosed find a sample category listing. Add glossary terms as needed).
2. Add heading to the paragraphs in several sections. This will allow better classification of material in guidelines.
3. Prepare a better table of contents.
4. Make more readable and logical wherever possible.

Distributed the graphics information from W. Loper. If the members are interested, please respond. If not, try to give information to someone else in your organization that is interested in graphics.

Future topics of investigation or study by this group were discussed. They were:

1. The possibility of proposing some form of standard presentation of graphic displays at the operators console.
2. Could we investigate the possibility of a standard operators console.

R. F. Ca

R. F. Ca

PRECEDING PAGE BLANK-not

APPENDIX A-V (Cont.)

SEND ONLY MINUTES:

R. F. Carroll
B.F. Goodrich Chemical Div.
6100 Oak Tree Boulevard
Cleveland, Ohio 44131
(216-524-0200)

Joseph C. Cermak
Bailey Meter Company
Mail Station 2E8
Wickliffe, Ohio 44092
(216-943-5500) (2341)

James P. Foster
Hershey Foods Corp.
19 E. Chocolate Ave.
Hershey, PA 17033
(717-273-8668)

Shoji Fukumoto
Fischer & Porter Co.
County Line Road
Warminster, PA 18974
(215-674-8492)

Alex Habib
Olin Corp.
120 Long Ridge Road
Stamford, CT 06904
(203-356-2719)

Robert L. Hartway
University of California
P. O. Box 1663 - M.S. 569
Los Alamos, New Mexico 87545
(505-667-7990)

Stephen P. Higgins Jr.
Honeywell Process Control Div.
2222 W. Peoria Ave.
Phoenix, AZ 85029
(602-943-2341)

Donald G. Kempfer
Standard Oil Company
Midland Building
Cleveland, Ohio 44115
(216-575-5516)

V. A. Lauher
Monsanto
800 N. Lindberg
St. Louis, MO 63011
(314-694-6506)

Fred J. Martino
Macro Corporation
1260 Virginia Drive
Ft. Washington, PA 19034
(215-646-9510)

Tom McLeod
Gould-Modicon
Box 83 SVS
Andover, MA 01810
(617-475-4700)

Remsi Messare
Exxon Research & Engrg. Co.
Box 101
Florham Park, N.J. 07932
(201-474-1374)

George T. Paulissen
Shell Oil Co.
Major Projects
Deer Park, Texas 75536
(713-476-6196)

Robert F. Sapita
The Foxboro Co.
Foxboro, MA 02035
(617-543-8750) (2233)

Richard D. Thompson
Applied Automation Inc.
201 RB 2
Barlesville, OK 74004
(918-661-3637)

Charles W. Watson
Leeds & Northrup
Dickerson Road
N. Wales, PA 19454
(215-643-2000) (8227)

Robert A. Williamson
Metromation Inc.
1101 State Road
Princeton, NJ 08648
(609-924-3900)

John Zarrella
Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051
(408-987-7432)

Paul M. Augenstein
Taylor Instruments
Sybron Division
95 Ames Street
Rochester, N. Y. 14601
(716-271-2220)

Tom Parris
Johnson Controls Inc.
507 East Michigan
Milwaukee, WI 53201
(414-276-9200) (588)

Lyle L. Simon
Cummins Engine Co.
1000 5th Street
Columbus, IN 47201
(812-379-6710)

Theodore J. Williams, Wkshp. Chrm.
Purdue Lab. for Applied Industrial Control
102 Michael Golden
Purdue University
West Lafayette, IN 47907
(371-494-8425)

APPENDIX A-V (Cont.)

REASONS FOR IMPLEMENTING AND NOT IMPLEMENTING MODERN MAN-MACHINE INTERFACE FUNCTIONS

The following list describes positive and negative factors which will influence any decisions regarding implementation of modern man-machine interface functions. The designer should carefully consider the items in these lists (as they apply to his system) and balance the positive and negative factors.

Positive Factors

The list of reasons for implementing a modern man-machine interface is divided into two groups. The first group contains those reasons which are generally applicable to any MMIF upgrade or new design. Those in the second group will usually be positive factors, but must be evaluated in terms of the particular application.

I. General Factors

1. Improves Operator Efficiency

   1.1 Better process representation.

   1.2 Less operator error (saves process downtime and improves product quality).

   1.3 Data presentation capability (pattern recognition, operation by exception).

   1.4 Display/calculation of derived variables.

   1.5 Better utilization of computer capabilities (all capabilities will not be utilized if the MMIF is difficult for the operator to interact with).

2. Reduces overall size of the control room.

3. Reduces maintenance (fewer control and indicators).

4. Permits easier upgrade for process growth and change (adaptability and flexibility).

5. Enhances operators job enrichment.

6. Eases operator training (especially for new operators).

7. Permits replacement of equipment which is no longer in production.

8. Improves process optimization.

9. Provides better managerial capability.

10. Allows implementation of distributed architecture.

APPENDIX A-V (Cont.)

II. Specific Factors

1. Operations manpower may be reduced (greater number of loops or batch operations handled by each operator; powerful capabilities may permit upsets to be handled by a single operator).

2. Implementation using modern hardware and architectures may be more reliable and have higher availability than older designs (independent, redundant, or distributed processing; fewer active and/or higher reliability parts).

3. Installed cost of the MMIF may be lower (cabling costs reduced on large systems).

4. Operator morale may be improved (utilization of state-of-the-art equipment; managing computer plant control).

5. An upgrade to an existing control system may require a modern MMIF for optimal operation.

Negative Factors

The following list of negative factors must be analyzed by the design engineer for his MMIF system. These factors are usually quoted as reasons not to implement a modern MMIF. However, in many cases, the actual design is misunderstood, and these factors may not be valid. In addition, the severity of these factors will depend heavily on whether an existing MMIF is to be updated or a new control system is to be implemented (negative factors are generally more severe for MMIF upgrades).

1. Present personnel may not be capable of designing and implementing the MMIF for a new system. They may feel more secure with the existing design and implementation.

2. Proven technology can be used. This offers less risk and is more available and better understood.

3. Operations and maintenance people and practices may be different with a new design.

4. Design may be overkill for the application; too elegant and of little "real" use.

5. New MMIF may have negative effects on operating personnel and their jobs.

6. Design may increase the degree of abstract of the process.

7. Reliability will be decreased due to use of common hardware.

APPENDIX A-V (Cont.)

8. Government regulations prohibit or greatly restrict new MMIF design (especially government standards/ licensing).

9. Unions/operators may be concerned about job dislocations.

10. Control room/process environment may prohibit a new design.

11. Costs of a new MMIF

    11.1 May include additional design costs for design time, technology development, training, or staffing.

    11.2 May include additional maintenance costs. These costs will normally be lower; however, due to better packaging, less different types of consoles, hardware commonality, on-line diagnostics.

    11.3 May include additional training costs for maintenance people and operator retraining.

    11.4 May include additional hardware and software costs for a more complex CPU system and higher utilization of new technology.

    11.5 May include additional installation costs. These costs will normally be lower; however, due to better electrical interfaces (esp. cabling) and smaller physical size.

    11.6 Are not directly related to profits (the "business" of the corporation).

12. Modern MMIF design may require a more sophisticated and expensive computer control system than would otherwise be required.

APPENDIX A-V (Cont.)

MODIFY TUNING PARAMETERS

Command:	Lead Lag PID
Control Parameters:	Address Verification Feedback Mode (C, A, M, CAS) Value
Verification Feedback:	OK - Operation Started Error - No Verification Control Status Error Security Violation

APPENDIX A-V (Cont.)

CHANGE SETPOINT

Command:	Enter Ramp
Control Parameters:	Address Verification Feedback Needed Value (Increment Per Unit Time)
Verification Feedback:	OK - Operation Started Error - No Verification Control Status Error Security Violation

APPENDIX A-V (Cont.)

VALVE OPERATIONS/MOTOR OPERATIONS

Command:	Open Close % Stop	Start Stop Speed
Control Parameters:	Address Verification Feedback Needed (Timeout) (Rate)	
Verification Feedback:	OK - Operation Started Error - No Verification Operation Timeout Control Status Error Security Violation	

APPENDIX A-V (Cont.)

Glossary Term \ General Area	Human Factors	Design Oriented	Application Oriented	User Oriented	Originally MMIF Oriented
Aesthetics					
Alarm					
Analog					
Annunciator					
Arousal Threshold					
Backup					
Bulletin Board Effect					
Channel					
Clock Shop					
Cognition					
Communication					
Compatibility					
Console					
Control					
Control System					
Control Room					
Conversational Mode					
CRT Display					
Data					
Digital					
Display					
Emmulate					

APPENDIX A-V (Cont.)

Glossary Area \ General Area	Human Factors	Design Oriented	Application Oriented	User Oriented	Originally MMIF Oriented
Event					
Glyph					
Graphic					
Hard Copy					
Hierarchy					
Iconic Communication					
Information					
Interface					
Job					
Joy Stick					
Key Operator					
Loop					
Man/Machine Interface					
Man/Machine System					
Manual Operation					
Mode					
Noise					
Off-Line					
Operator					
Optimize					
Perception					
Plot					

APPENDIX A-V (Cont.)

General Area / Glossary Term	Human Factors	Design Oriented	Application Oriented	User Oriented	Generally MMIF Oriented
Population Sterotypes					
Process					
Rank					
Record					
Reliability					
Remote Stations					
Selection					
Sign					
System					
Task					
Task Analysis					
Task Description					
Task Synthesis					
Top Down					
Trending Recorder					
Trivia					
Transparent					

APPENDIX A-V (Cont.)

HUMAN FACTORS - (Human Models, long & short term memory). (Color blindness, response time).

DESIGN ORIENTED - Material that pertains to the technical specification of parts that comprise the subject in question.

APPLICATION ORIENTED - The use of designed materials (hard & soft) to build, assemble, or adapt, a MMIF system.

(Equipment selection, development of control room, implement a control scheme).

USER ORIENTED - The operation (use & maintenance) of a piece of MMIF gear to perform a process function.

APPENDIX A-VI

TC-7

RELIABILITY, SAFETY AND SECURITY COMMITTEE

APPENDIX A-VI

PURDUE WORKSHOP - TC-7 REPORT

First, I send my apologies to the Workshop for not being able to attend at this time.

Two meetings of the "Reliability, Safety and Security Committee", TC-7 were held since the Fall International Conference. They were primarily directed to the goal of stimulating interest in the purpose and work of the Committee and formulating a Committee operational strategy.

The first meeting held at the PPG offices in Pittsburgh in November 1977 was attended by representative from major industries.

C. Peterson	-	Columbia Gas
R. Hath	-	U. S. Steel
W. Brown	-	Celanese Corporation
R. Yunker	-	PPG Industries

The discussion centered on the purpose and direction of the committee. The activities of the European and Japanese Committees were discussed in particular the glossary of the European Commitee was reviewed. Experiences relating to reliability and availability were exchanged.

It was agreed that increased membership was vital to the committee's direction and as such the participant agreed to actively solicit and recommend possible candidates.

The second meeting also was held in Pittsburgh on April 4. Its objective was to continue to develop programs for increasing U. S. participation and in particular to study strategies being employed by attendees to improve system reliability. There was a very samll attendance. Discussion centered around Reliability Strategy of the represented companies. A paper prepared by R. W. Yunker on strategies will be distributed to members.

Plans are to actively solicit nuclear power company and vendor participation to increase participation to a minimum of ten(10) people.

The next meeting is scheduled for Pittsburgh on August 1, 1978, in the PPG offices, One Gateway Center.

Please contact R. W. Yunker - (412) 434-3377 - for further information.

APPENDIX A-VII

USA TAG

ISO/TC97/SC5/WG1

PROGRAMMING LANGUAGES FOR INDUSTRIAL PROCESSING

ansi american national standards institute, inc
1430 broadway, new york, n.y 10018 (212) 868-1220

APPENDIX A-VII

MAY - 4 1978

April 25, 1978
Scott Paper Company
Scott Plaza III
Philadelphia, PA 19113

Minutes of the USA TAG Meeting for ISO/TC97/SC5/WG1 held at Purdue University on Monday, April 10, 1978.

A meeting of the USA TAG for ISO/TC97/SC5/WG1 was held at Purdue in conjunction with the meeting of the American Region of the International Purdue Workshop. Dr. M. R. Gordon-Clark and Mr. M. E. Adams reported on the Working Group meeting held in London in November 1977. Dr. T. J. Harrison and Lt.-Col. W. A. Whittaker were unable to be present. Twenty eight people were present (see attached list).

Dr. Gordon-Clark reviewed the salient results of the London meeting in three areas, functional requirements, Industrial FORTRAN and CORAL-66. The functional requirements documents prepared by the British and German experts were reviewed and minor changes made. In Industrial FORTRAN, the working group agreed to submit to SC5 for a letter ballot ANS/ISA S61·1-1976 with the addition of some extra comments explaining the limited extent of the executive interface calls and clarifying some technical definitions. The standard ISA S61·2-1978 was discussed in detail and the differences with the European Real-Time FORTRAN (Primarily a German document) were mentioned. The working group requested the US and German experts prepare a document by the next working group which was satisfactory to all experts. A proposed solution was developed at the Purdue Europe meeting but the details have still to be worked out. The discussions on CORAL-66 did not resolve the issue of the importance of imbedded real-time features but it was agreed to recommend to SC5 that CORAL-66 be standarized by SC5 as a general language of greater range than that covered by the scope of the working group.

Mr. Adams attended the meeting of ISO/TC97/SC5 at the Hague in the Netherlands. The two resolutions of WG1, namely on ISA S61·1 and on CORAL-66 were approved. The SC5 committee will send out for letter ballot ANS/ISA S61·1 when it is received by the SC5 secretariat and the British Standards Institute (BSI) were asked to prepare CORAL-66 for submission to SC5. In addition Mr. Adams discussed the action of SC5 on FORTRAN, PL/1 and BASIC.

It was agreed to hold the next meeting of the USA TAG at the October meeting of the International Purdue Workshop.

M. R. Gordon-Clark

PRECEDING PAGE BLANK - NOT FILMED

APPENDIX A-VII (Cont.)

April 10, 1978

Meeting of US TAG to ISO TC97/SC5/WG1-PLIP

Name	Company/Organization	Vendor User University
Merritt Adams	Western Electric	User
M. R. Gordon-Clark	Scott Paper	User
A. J. Arthur	IBM Corporation	Vendor
Gary L. Troter	Atlantic Richfield	User
Jeffrey K. Fang	Atlantic Richfield	User
Carl Wilson	Texas Instruments	User/Vendor
Mark S. Borowiak	Inland Steel	User
R. F. Thomas, Jr.	Lawrence Berkeley Lab	User
R. D. Hawkins	Naval Weapon Center	User
W. E. Loper	Naval Ocean Systems Center	User
Donald B. DeVorkin	Computer Sciences Corporation	User/Vendor
Stephen C. Schwarm	E.I. duPont deNemours Company	User
John D. Higham	Measurex Corporation	User/Vendor
R. M. Lee	Measurex Corporation	User/Vendor
V. A. Lauher	Monsanto	User
K. E. Platt	Bethlehem Steel	User
S. L. Gifford	Bethlehem Steel	User
Robert G. Wilhelm, Jr.	Industrial Nucleonics Corporation	User/Vendor
Shoji Fukumoto	Fischer & Porter Company	User
Alex Habib	Olin Corporation (Chemicals)	User
D. L. Ness	Cutler-Hammer	User

APPENDIX A-VII (Cont.)

Name	Company/Organization	Vendor User University
Robert F. Carroll	B. F. Goodrich Chemical	User
Stephen G. Hussar	PPG Industries, Inc.	User
Richard C. Flata	Leeds and Northrup	Vendor
Ed Rathje	Fischer & Porter	Vendor
Richard W. Signor	General Electric Company	User
Richard H. Caro	The Foxboro Company	Vendor
Bruce Stowell	Hewlett-Packard	Vendor

SECTION IV

TECHNICAL APPENDICES

APPENDICES E-VII

TC-1

REPORT OF THE INDUSTRIAL REAL-TIME
FORTRAN COMMITTEE, PURDUE EUROPE

1. Industrial Real-Time FORTRAN, Definition of Procedures for the Application of FORTRAN for the Control of Industrial Processes, January 1978.

PE TC 1 , 1/78 January 1978

Industrial Real-Time FORTRAN

Definition of Procedures for the Application of FORTRAN for the Control of Industrial Processes

Proposal of the
Purdue Europe Technical Committee 1
on Industrial Real-Time FORTRAN

by

P. N. Clout	GB+D
A. J. Cox	GB
G. Heller	D
W. Kneis	D
W. Koblitz	A
K. M. Maliszewsky	P
O. Pettersen	N
U. Rembold	D
D. A. Rutherford	GB
G. Teuschler	D
M. Topschowsky	D
P. Urbainsky	D
G. Wiesner	D

The development of this proposal has been supported by the Commission of the European Communities, Directorate-General III

CONTENTS

Industrial Real-Time FORTRAN

1 INTRODUCTION

1.1 History and Development of Industrial RT-FORTRAN

The FORTRAN language was originally an IBM-development in 1955. Since that time FORTRAN has become the most widely used high level language for scientific applications, and large powerful user libraries are available in FORTRAN.

FORTRAN was proposed for standardization in 1967 and was finally standardized internationally in 1972 by ISO at three levels "Full FORTRAN", "Intermediate FORTRAN" and "Basic FORTRAN" [1].

The wide use and the proved excellence of FORTRAN for scientific applications soon led to its use for industrial real-time systems. These applications require special operations, i.e. real-time operations, bit-string manipulation and facilities for process-I/O. These operations can only be accomplished in one of the following two ways:

(1) FORTRAN remains the basic language for arithmetic calculations and for reading and writing of data by standard peripherals. The special operations are realized by elements outside the syntax of FORTRAN.

(2) The complete real-time language remains within the syntax of FORTRAN including the special operations. This means that these operations have to be FORTRAN-subroutines or FORTRAN-functions.

The first approach leads to typical real-time languages which offer the user simple but powerful programming facilities for the special operations. In developing such extensions the designers try to apply all the features of the real-time operating system used; thus these extensions become dependent on the actual system. This was especially true for the early developments of this kind (see e.g. [2], [3], [4]). PROCOL, as a later development, avoids this disadvantage; this language was created in France for a series of French computers. It offers very advanced features for real-time programming (see e.g. [5]).

All languages with extensions outside the syntax of FORTRAN need special compilers for their translation.

In choosing the approach (2), where the language remains within the syntactical frame of FORTRAN, one gains the advantage that a first compilation and check of the user program can be done on any computer for which a FORTRAN compiler exists. On the other hand this approach, using subroutines and functions, leads to somewhat clumsy handling of the added CALLs and FUNCTIONs and their associated parameters. Yet this may be considered as a minor inconvenience as the many users of FORTRAN are well accustomed to this kind of programming.

Industrial real-time FORTRAN has also to be compared with two other families of real-time languages: industrial real-time BASIC and real-time languages of the LTPL-type like PEARL.

The industrial real-time BASIC languages are very easy to learn and to apply. They are well suited for simple and small problems. They can be implemented relatively easily in large as well as in small computer systems.

The languages of the LTPL-type deliver very powerful programming features to the user. These languages are therefore well suited to large and complex problems. Their implementation (compiler and real-time system) is relatively expensive.

The industrial real-time FORTRAN languages implemented by approach (1) above are often similar to the LTPL-type languages. On the other hand the industrial real-time FORTRAN languages implemented by approach (2) are in many respects between BASIC and LTPL-languages. Thus this language offers the user an alternative to these two language facilities. Consequently, a language according to approach (2) has to be relative simple; this means that the number and complexity of the additional operations have to remain restricted because of the ease of learning and programming.

In order to prevent the development of many incompatible real-time languages, T. J. Williams and others in 1970 founded the "Workshop on Standardization of Industrial Computer Languages" at Purdue University [1]). The "FORTRAN Committee" of the Purdue Workshop was very active and successful from the beginning. For various and good reasons this committee chose approach (2) above with all special operations being kept within the syntactical frame of the FORTRAN language. The committee developed in a relative short time a first proposal for real-time FORTRAN which was approved and published by the ISA as ISA Standard S61.1 (1972) [6]. The paper contains the following groups and numbers of special operations:

- 3 CALLs and the FORTRAN statement STOP for controlling the state of parallelly activated "programs" [2]).
- 6 CALLs for process-I/O
- 5 INTEGER FUNCTIONs for bit-string manipulation applied on an INTEGER used as a bit-string.

---

1) *After a union with another institution in* 1973, the workshop was named "International Purdue Workshop on Industrial Computer Systems".

2) In this paper the term "activity" will be used instead of the term "program" used in [6], [7], [8], and [9]. See section 1.2. This is in accordance with current terminology at Purdue Workshop.

A second supplementary paper ISA S61.2 (1973) was published one year later. This paper contains the following groups and numbers of special operations:

- 7 CALLs for handling random unformatted files
- 1 LOGICAL FUNCTION for testing an individual bit in an INTEGER
- 2 CALLs for setting and clearing of an individual bit in an INTEGER

These ISA-standards have gained great acknowledgement and, by worldwide use, great importance (the proposal for file handling excepted). The early developement and the publication of these standards by the American "FORTRAN Committee" of the Purdue Workshop and the ISA has been an important step for industrial process control.

In the meantime the ISA standards have been improved in ISA S61.1 (1976) [8] which now contains all sections of the old papers besides the file handling; the file handling is now described in draft ISA S61.2 (1977) [9]. The major change is that CALLs which do not wait on the completion of the procedure are no longer allowed. Besides this there are only minor differences.

As the paper ISA S61.1 now contains all CALLs and FUNCTIONs which have gained wide acceptance, this paper can be considered to be the "Basic Industrial Real Time FORTRAN" of the Purdue Workshop. It contains only 3 CALLs and the statement STOP for the management of parallel activities (= programs). As a result the American "FORTRAN Committee" has worked very intensively on a supplementary paper about the management of parallel activities which will be published as ISA S61.3. Parallel to this work, the working group "Prozess-FORTRAN" of the German VDI/VDE has developed a complete proposal "Prozess-FORTRAN 75" [10], which will be published as a draft standard of the VDI/VDE in the near future.

"Prozess-FORTRAN 75" comprises 11 CALLs for the management of parallel activities and thus offers a restricted but powerful tool for programming real-time operations. The binary pattern and bit processing is very similar to ISA S61.1 (1976); merely the description is different. There are additional INTEGER FUNCTIONs for arithmetic and circular shift and a CALL for a bit change. The process I/O is practically identical to ISA S61.1 (1976); only the standardization of the analog I/O is performed one step further. The file handling is different from the proposed ISA S61.2 (1977).

In 1976 the Purdue Europe Technical Committee 1 on Industrial Real-time FORTRAN was founded by members of the VDI/VDE working group "Prozess-FORTRAN" and other specialists in Europe. The committee decided to work out a complete proposal for industrial real-time FORTRAN. As already mentioned, the American "FORTRAN Committee" has developed and modified existing work by publishing papers supplementary to existing ISA papers; for them this is certainly the best way to proceed. For the Europeans to work in this way would require a very close cooperation with the American "FORTRAN Committee"; but this is not practical in the present situation with only one joint meeting per year. Thus cooperation must be by the exchange of papers. The European Technical Committee 1 thinks that working out a complete proposal is the best way to exchange ideas. It must be emphasized that this paper is not a competing product to the American papers: in the present situation it is meant to be a means of showing the ideas of Purdue Europe Technical Committee 1. It must also be emphasized that this paper is identical largely to and influenced strongly by the American papers.

In the following, a short introduction and overview of the sections of this paper is given.

In section 1.2 the definition of important terms used in the paper is presented.

The most difficult and controversal part of industrial real-time FORTRAN is the management of parallel activities handled in section 2. For this section the committee used the paper [11] of O. Pettersen as a basis. In this very detailed paper, besides the CALLs already described in [10], semaphores and conditional critical regions are proposed for the synchronization of parallel activities. The committee decided not to include the conditional critical region in the proposal as this feature is not currently available in real-time operating systems of general knowledge. Compared to the solutions of LTPL-type languages, the management of parallel activities proposed here gives the user limited power to influence foreign parallel activities but complete control of all CALLs by means of associated failure parameters. Thus programming can be made quite secure.

Section 3 on binary pattern and bit processing and section 4 on process input/output are completely identical to the "Prozess-FORTRAN 75" [10], except from editorial details.

Section 5 on file handling is very similar to the "Prozess-FORTRAN 75" [10] proposal. In this section programming and file security are the main issues considered.

This Industrial Real-Time FORTRAN standard will need to be adapted from time to time to allow the progress of FORTRAN (see for example [11]) and also the progress of real-time operating systems. Thus supplements must be expected.

1.2. Definitions of important terms

Section 1.2 should be referenced when reading this paper but may be omitted from the first reading.

The following terms and their definitions are used throughout this document. Some of them are inspired by or borrowed from literature as [13], [14], and other sources. Most of the definitions correspond largely to those commonly used at International Purdue Workshop and have been presented in an earlier committee document [11]. Definitions of the states refer to Figure 1 of Section 2 and its description in Section 2.2 and may be clearer if the figure and its description are consulted.

The terms are written in capital letters in their definitions. Terms defined elsewhere in this vocabulary are underlined. It will, however, be underlined only at the first occurrence within the same definition paragraph.

ACTIVITY	A <u>computation</u>, where the <u>operations</u> are performed in a strict sequential order.
CALLING ACTIVITY	The <u>running activity</u>, in whose <u>program</u> the executed subroutine call statement is contained.
CALLING PROGRAM	The <u>program</u> which contains the executed subroutine call statement. See definition for <u>calling activity</u>.

COMPUTATION A finite set of operations applied on a finite set of data, in an attempt to solve a problem. If the computation really solves the problem, it will also be an ALGORITHM.

CRITICAL REGION A part of a sequential program operating on shared data such that this program part must have exclusive access to the shared data during the execution.

DORMANT One of the definite states of an activity. A dormant activity is known to the executive system and is not in any of the states pending, running, or suspended.

EVENT A significant discrete occurrence or incident which is intended to affect some program execution in a planned manner. The source of an event can belong logically to an entity distinctively apart from the affected program unit or units. An event itself occurs instantaneously and is of infinitesimal duration. The fact that an event has occurred is indicated by an EVENTMARK, which is a formal boolean entity.

The occurrence of an event may be external or internal: The effect of an external event will be transmitted through the physical processor's input interface system as an interrupt request signal, or a signal actively read by some program statements. The source of an external event will generally be of some physical nature. An internal event is characterized by some condition change within a program as a consequence of some program action. With relation to the effect of an event, it is not distinguished between external or internal nature of its origin.

As evident from the definition, time can be a basis for an event. For example, a prevalent time is defined, relating to the equation:

$$B \leftarrow t > t1$$

where:

t is wall clock time
t1 is some predetermined point of time
B is the eventmark, recognized as a binary boolean entity.

EXECUTION The collection of actions performed by a computer processor carrying out instructions in a sequential manner.

EXECUTIVE SYSTEM The part of an operating system controlling the allocation of resources in a computing system. One major resource in this respect is processor execution time. the executive system maps a virtual processor to a physical processor.

INTERACTIVE SYSTEM A computer with operating system allowing computations to interact with the computer's environment or surrounding world. Interactions may or may not be intentional.

MULTIPROGRAMMING Programming techniques or organizational forms which make use of, or allow, parallel activities.

NAME OF ACTIVITY A FORTRAN symbolic name.

NON-EXISTENT One of the definite (formal) states of an activity. A non-existing activity is unknown to the executive system.

OBJECT ACTIVITY The activity that is wanted or expected to be started, halted, stopped, or otherwise affected as a consequence of a system subroutine call. It may also be termed the DESIGNATED ACTIVITY.

OBJECT PROGRAM Since all subroutine calls described here relate to object activities, rather than programs, the term "object program" will rarely be used. Where applicable, however, this term means the program, or a program, used by an activity under its execution.

OPERATING SYSTEM An organized collection of system programs acting as an interface between computer hardware and users or user's programs, providing the latter with a set of facilities to simplify the design, coding, program error discovery, and maintenance of programs, as well as controlling the allocation of resources to assure efficient operation. See executive system.

OPERATION A deterministic rule for the generation of a finite set of data from another finite set of data.

PARALLEL ACTIVITIES A set of activities whose operations may overlap in time. Parallel activities are DISJOINT if every one of them only refers PRIVATE DATA, i.e. data that are not accessed by any other activity of the set. They are INTERACTING if they refer to SHARED DATA, i.e. data that are accessed by more than one activity of the set.

PENDING One of the definite states of an activity. A pending activity has been associated with an event by another activity such that when the event occurs, the activity will be transferred to state running and start its execution from the main entry-point (top) in its program.

PHYSICAL PROCESSOR A physical component capable of executing an activity defined by a stored program.

PRIORITY A number used to establish an order of precedence among activities competing for resources.

Priorities can be fixed or dynamic, and a certain activity may have different priorities related to different types of resources.

PROGRAM A description of a computation, expressed in a formal language, PROGRAMMING LANGUAGE.

RUNNING One of the definite states of an activity. A running activity is executing in its virtual processor. This means that its program will be executing on a physical processor if all necessary resources are assigned to it. Otherwise, it will be waiting for its required resources.

SEMAPHORE A structure of shared data, used for the exchange of synchronizing signals between interacting parallel activities. Operations on a semaphore must only be performed within a critical region.

SEQUENTIAL PROGRAM A program consisting of operations that can only be performed strictly one at a time without any overlap in time. See definition of activity. An activity is described by one or more sequential programs excluding each other in time.

SUSPENDED One of the definite states of an activity. A suspended activity has temporarily halted the execution of its virtual processor, and is waiting for a specified event to continue the execution of its virtual processor. It is emphasized, that a suspended activity has halted in the midst of its execution and will later resume its execution at the point where it was interrupted. This is in contrast to a pending activity, that will always start at its main entry point.

SYNCHRONIZATION A general term for restrictions regarding the order in which operations are performed. E.g. a synchronization rule may specify the order, priority, or mutual exclusion in time between two or more operations.
Specifically, the term synchronization will here be tied to coordination of parallel activities. According to the definition above, only interacting parallel activities are relevant to consider for synchronization.

VIRTUAL PROCESSOR A simulated component, capable of executing an activity defined by a stored program. By duty of the executive system, the virtual processor may be realized by a physical processor or simulated by program execution within the executive system.

2 MULTIPROGRAMMING AND REAL-TIME FEATURES

This chapter describes several subroutine calls available for the user's program, and relating to multiprogramming and particularly real-time operation. For all calls of chapter 2 the operation is generally indivisable, i.e. the operation of these calls shall not be interrupted.

2.1. Date and Time information

For programming in a real-time environment, the user must have access to the time variables of the operating system. These time variables are obtained by system calls described in this section.

Unambiguous time specification requires non-modular designation of time including complete date and an acknowledged calendar, defining time zero. Execution of reference to subroutine DATIM provides this complete information, comprising the information obtained by two subroutine calls of ISA - S61.1. The date refers to the Gregorian Calendar.

The calls are:

CALL DATIM(j) For obtaining current date and time.

CALL CLOCK(j) For obtaining the basic counts of the system's real-time clock.

2.1.1. Obtain Date and Time

The form of this call is:

CALL DATIM(j)

where:

j Designates an integer array, into whose first 7 elements will be placed the absolute time, as expressed by the systems's real-time clock at the time when the call is executed. These elements are as follows:

First element:		Year (1977 →)
Second	"	Month (1 to 12)
Third	"	Day (1 to 31)
Fourth	"	Hours (0 to 23)
Fifth	"	Minutes (0 to 59)
Sixth	"	Seconds (0 to 59)
Seventh	"	Milliseconds (0 to 999)

2.1.2. Obtain Clock Counts

Execution of a reference to this subroutine allows the determination of the basic counts of the system's real-time clock. The origin of the clock is arbitrary. The form of this call is:

CALL CLOCK(j,k1,k2)

where:

j Designates an integer variable containing the basic counts of the system's real-time

clock as a positive integer. j is counted up to the maximum and is then assumed to be changed to zero.

k1 Frequency of clock in cycles per second. An integer constant returned by the system.

k2 Specifies the half period of the clock in basic counts. An integer constant returned by the system.

2.2. A mathematical model for parallel activities

2.2.1. States and transitions

An activity may be described as a "sequential machine" of Moore type. At any time, the activity is in one and only one state. Actions executed by the operating system, other activities, or the designated activity itself, may cause transition from one state to another. These transitions are performed instantly, i.e. they are considered ideally to take no time.

This mathematical model of an activity may be visualized by a "state diagram", like Fig. 1, in which the states are nodes, illustrated as circles, while transitions are drawn as pointed arrows from one node to another. [11].

A multiprogramming system, which per definition consists of several parallel activities, can be considered as modelled by a number of disjoint but similar diagrams. It is feasible to apply a three-dimensional picture, with the similar diagrams sandwiched on top of each other and oriented so that the identical states of the individual diagrams cover each other. The total system will have a number of state "stacks", the same number as the number of states in the diagram. Each "stack" represents the particular state of all activities, and it is easy to impose certain restrictions to the maximum number of activities entering the same state: In the composite model, this will be the maximum number of "entered" wafers in the stack. For the system, the limitations are dictated by available resources, which, for example, can be table space.

State transitions are generally caused by subroutine calls; the name, form, and interpretation of which are standardized and described in the present document. Additionally, in one special case, a transition is caused by a standard FORTRAN statement: STOP.

In the present document, an activity is considered as described by a mathematical model illustrated by the state diagram of Fig. 1. This model adheres to the following basic principles:

1. Transitions are non-ambiguous, i.e. for a given stimulus in a given state, the activity can transit to only one possible new state.
2. Transitions are performed instantly, i.e. in zero time.
3. The model is a Markov system, i.e. transitions from a state depend only upon the state and current conditions, not on the previous state and the conditions that brought the activity into the present state.
4. An activity exists in one state only at a time.

The model used in this document describes the behavior of activities as seen from the application programmer.

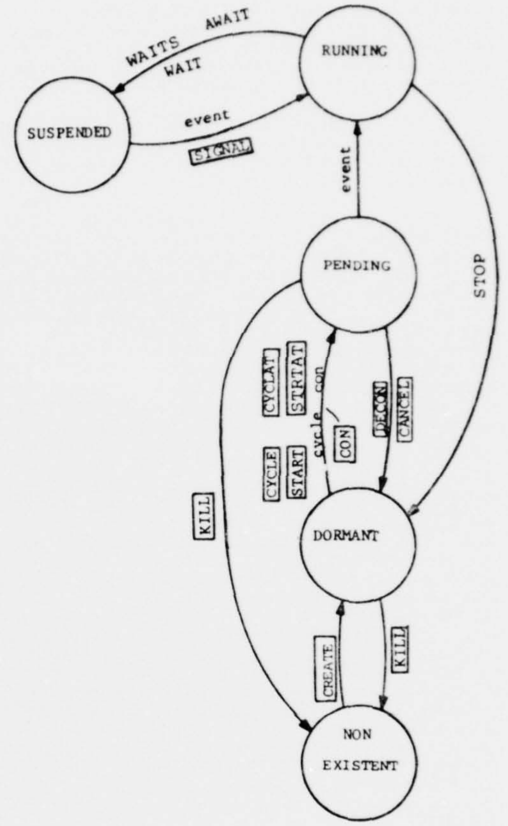


Figure 1. STATE AND TRANSITION DIAGRAM

No attempt is made to describe operating system actions transparent to the application programmer, and not directly intentional of the application program. Thus, the state RUNNING is related to the activity's virtual processor: It is immaterial for the state of an activity, whether a physical processor happens to be assigned to it, or the execution is temporarily hampered by the operating system due to limited availability of physical processors and the activity's low priority.

The following symbolism is used for transitions in the state diagram:

- Capital letters in box: Effect on an activity imposed by another activity. I.e. a subroutine call in one activity has the indicated effect on another activity.
- Capital letters without box: Effect imposed by the activity itself while in the state RUNNING.
- Small letters: Conditions under which the Executive system performs the indicated state transitions.

2.2.2. Multiple calls for state transitions

Different activities may issue conflicting transition calls regarding the same object activity. This will be the case during normal operation, as well as under error conditions, since the state of an object activity is indeterminate at the time a transition call is made by another activity.

In order to adhere to the requirement of one state only at a time, as listed in the previous paragraph, the problem of conflicting transition calls is solved as follows:

Call to the executive system (by calls of START, STRTAT or CON) for state transition to PENDING causes listing in a table, appropriately called "pending-table". Each activity may have a maximum of one entry in this table, but complex conditions are allowed: Limited only by some maximum table size, all valid subroutine calls intended to cause state transition to PENDING should have its "running-condition" OR-ed to the event-function constituting the condition for the subsequent state transfer to RUNNING. This rule gives a simple and logical operation of the CYCLE, CYCLAT, and CON calls, i.e. the calls causing possibility of multiple subsequent activations (RUNs): Whenever an activity attains state DORMANT, (by execution of STOP), the pending-table is checked. If listed, the activity will be transferred immediately to state PENDING, guided by an OR-function of run-conditions from the table-entry. If the listing call has one of the recurrence calls CYCLE, CYCLAT, or CON, the table condition will remain. The same will be the case, if multiple calls of START or STRTAT were made, leaving the future activation times in the table. Thus, there will be no difference, in principle, between the immediate effects of multiple calls of START or STRTAT and the recurrence calls CYCLE, CYCLAT, and CON. The effect of the recurrence calls will only be, that the remaining conditions in the table entry will be adjusted recursively, at the instant when the activity after a STOP is transferred past DORMANT to PENDING. The minimum size of the "pending-table" shall allow one extra call to be buffered, i.e. one call if the activity is in one of the active states (RUNNING or SUSPENDED).

A different issue is the problem caused by activities ending their execution excessively delayed, so that the time or other RUNNING conditions for their next execution, as scheduled by call of START, STRTAT, CYCLE, CYCLAT, or CON is already satisfied before STOP of the previous run is executed. Such conditions should normally be regarded as errors and give some error reaction. The problem is, however, that the error condition does not exist when the scheduling call is made, and most systems do not keep any record of connections back to the scheduling activity; at least, such operations would hardly conform to FORTRAN. Thus, there is no obvious receiver of such error reactions within the user's program. Most appropriately, such error reactions should be handled by the system, and as such, it is outside the scope of this standard.

2.3. Subroutine and procedure calls

2.3.1. Terms and Summary of subroutine calls

This paragraph contains a summary of the subroutine calls described in subsequent paragraphs.

The following terms, used in the description of the calls, are defined in sec. 1.2.: "calling activity", "calling program", "object activity", "object program", and "name of activity".

The following designations for parameters apply to several of the calls. If the exact meaning of these parameter designations deviates from what is described below, it will be marked specifically in the detailed description of the call. If the meaning is exactly as defined here, the description of the parameter is omitted in the description of the call.

i — specifies the activity to be affected (object activity). The argument shall be an array name.

m — is set on return to the calling program, to indicate the disposition of the request as follows:

0 or less	:	undefined
1	:	request accepted
2 or greater	:	request rejected

This argument shall be an integer variable or integer array element, local to the calling program.

The list of function and subroutine calls, described in detail in subsequent paragraphs, is:

Full description in paragraph:	call:	parameters:
2.3.2.	CALL CREATE(i,j,m)	i : name of created activity j : descriptor of associated program
2.3.3.	CALL KILL(i,j,k,m) (opposite of CREATE)	j : termination parameters
2.3.4.	CALL START(i,j,k,m) (relative time)	j : numerical value of time delay k : time unit
2.3.5.	CALL STRTAT(i,j,m) (absolute time)	j : reference to time descriptor for activation instant
2.3.6.1.	CALL CYCLE(i,j,k,l,m) (cyclic, with delayed first activation)	j : length of time interval k : time unit l : time delay before first activation

2.3.6.2.	CALL CYCLAT(i,j,k,l,m) (cyclic, with absolute time spec. of first activation)	i,j,k,m : identical to parameters of CYCLE l : reference to absolute time descriptor for activation instant
2.3.7.	CALL CON(i,e,m)	e : reference to event descriptor
2.3.8.	CALL DECON(i,e,m) (eliminate event connection)	e : like CON
2.3.9.	CALL CANCEL(i,m) (eliminate scheduling)	
2.3.10.	CALL WAIT(j,k,m) (delay continuation of calling activity)	j : length of time delay k : time unit
2.3.11.1.	CALL WAITS(r,j,m) (wait on semaphore)	r : semaphore reference j : decrement
2.3.11.2.	CALL SIGNAL(r,j,m) (release semaphore)	r : semaphore reference j : increment
2.3.11.3.	CALL PRESEM(r,s,m) (initialization of semaphore)	r : semaphore reference s : initial value of semaphore
2.3.11.4.	RDSEM(r,m) (read semaphore value)	r : semaphore reference function value: value of semaphore variable
2.3.11.5.	CALL AWAIT(e,j,k,m)	e : event specification for resumed execution j : numerical value of time specification k : time unit
2.3.12.	STOP (termination of execution).	

2.3.2. Creation of a new activity

A new activity is introduced to the real-time system by reference to subroutine CREATE. The designated activity will be associated with some specified program, considered a resource like other resources, necessary for the activity to perform. The associated program is normally assumed to exist in an executable form. Specifically, linking is not a part of the operations under this call. (I.e. linking is supposed to have been done prior to this call.)[1)]

The form of the call is:

CALL CREATE(i,j,m)

where:

i — is an output parameter whose contents are supplied by the system. The argument specifies the created object activity and is an array name.

j — specifies an integer array which contains all information necessary to specify an associated program: its designation, where the program can be found such as description of file, residency while existent (primary memory resident or swappable), etc. This array will also contain the activity's processor priority. The details are processor-defined.

m — see 2.3.1.

2.3.3. Eliminating an activity from the real-time system

A reference to subroutine KILL will eliminate a designated activity from the real-time system, by transferring it to state NON-EXISTENT. If the designated activity is in state DORMANT or PENDING, the effect shall be carried out immediately. If the designated activity is in any executing state, the termination shall only affect future executions. Thus, the designated activity will conclude its present execution, without any intervention by this call.

The processor may impose certain restrictions regarding which calling activities may be permitted to eliminate other activities. For example, a reference to subroutine KILL may be effective only if the designated activity has been created by the same calling activity, i.e. there exists a parent - child relationship.

The form of the call is:

CALL KILL(i,j,k,m)

where:

i,j — see 2.3.2.

---

1) It is assumed to exist a mechanism outside the standard, to create the first activity, i.e. the "father", which in its turn will create other activities.

k specifies an integer array, an input parameter which contains information to guide the termination, such as whether the object program is to be written to some background storage, the location of the file within such storage, etc. The specific details are processor-defined.

m is set on return to the calling program, to indicate the disposition of the request as follows:

0 or less	: undefined
1	: request accepted
2	: request accepted, but the activity is still executing within a run already started prior to the call.
3 or greater	: request rejected

This argument shall be an integer variable or integer array element, local to the calling program.

2.3.4. Starting an Activity Immediately or after a Specified Time Delay.

Execution of a reference to the subroutine START shall, after the expiration of the specified time delay, cause the execution of the designated activity's first program unit. This execution shall commence at the program's first executable statement. The time delay is defined as the nominal duration from the time the call is made until the time the designated activity is supposed to enter state RUNNING. In the meantime, the designated activity is listed in a table, and is tranferred to state PENDING if the present state is, or when it becomes, DORMANT. The actual time instants for the entering and the leaving of state PENDING are subject to the resolution of the system's real time clock, to the interrogating and activating actions performed by the Executive system, and to the possibility that the present state is not DORMANT. Thus, a reference to subroutine START is not necessarily equivalent to a transfer to state PENDING, but to a request for such a transfer, when it becomes permissible. A specified zero delay shall be interpreted as "Immediate execution", which shall cause any residency in state PENDING to vanish or be as short as possible. The call shall be rejected, with m=2, if a negative delay is specified.
The form of the call is:

```
CALL START(i,j,k,m)
```

where:

i,m see 2,3.1.

j specifies the time delay, in units as specified by k, and as defined above. This argument shall be an integer expression.

k specifies the units of time as follows:
0 - Basic counts of the system's real time clock
1 - Milliseconds
2 - Seconds
3 - Minutes
This argument shall be an integer expression.

2.3.5. Starting an Activity at a Specified Absolute Time.

Execution of a reference to the subroutine STRTAT shall cause the designated (object) activity's first program unit to be executed at a specified time (absolute time). Execution of the object program will commence at the program's first executable statement. The specified time shall be the time when the object activity is supposed to enter state RUNNING, and the actual time of this occurrence is subject to the resolution of the system's real time clock and to the interrogating and activating actions performed by the Executive system. The relations to transitions between states DORMANT, PENDING, and RUNNING are similar to those described for subroutine START. The call shall be rejected, with m=2, if the specified time has already occurred, i.e. if the call is executed later than the instant when the object activity was supposed to start its execution.
The form of this call is:

```
CALL STRTAT(i,j,m)
```

where:

i,m see 2.3.1.

j Designates an integer array, whose first 7 elements contain the absolute time at which the object program is to be executed. These elements are as follows:

First element:	Year (0 ⟶)
Second "	Month (0 to 12)
Third "	Day (0 to 31)
Fourth "	Hours (0 to 23)
Fifth "	Minutes (0 to 59)
Sixth "	Seconds (0 to 59)
Seventh "	Milliseconds (0 to 999)

As a feature of convenience, value zero in one of the three date elements designates "current date". It is emphasized, however, that this should only be considered a matter of convenience, available for the user programmer. The input procedures shall immediately change "0" to current values, before any table entries are made, and it is the user's responsibility to make sure, that a wrong date will not be the result, according to what is explained above.

Any value outside the ranges specified shall result in an error return.

This argument shall be an integer array name.

2.3.6. Starting an Activity in Periodic Execution.

2.3.6.1. Initial start immediately or after a specified time delay.

Execution of a reference to the subroutine CYCLE has the same immediate effect as a reference to the subroutine START. Additionally, the designated activity shall be re-scheduled each time the activity leaves state PENDING. The new time scheduled shall be equal to the sum of previous scheduled time and the interval specified by the call of CYCLE. The re-scheduling under said condition will continue until actively terminated by a call of subroutine CANCEL,

thus stopping a series of periodic executions. After such a termination, a new cyclic execution of the object activity will require new call of CYCLE. The interval can be modified by another call of CYCLE, without an intervening call of CANCEL. This modification shall, however, affect only succeeding scheduling operations, i.e. it shall not change the current interval and the execution which is already scheduled; the "delay" parameter of a modifying CYCLE call is without effect. If the "delay" parameter shall have effect, the previous cycling must first be terminated by a reference to subroutine CANCEL.

The actual running may be delayed unintentionally, while in state RUNNING, because of running of other programs. Such delays will not be accumulated. If the execution is not finished before the time for next execution, the next execution will be cancelled and a system dependent action should be generated.
The form of the call is:

```
CALL CYCLE(i,j,k,l,m)
```

where:

- i — specifies the activity to be executed. The argument is either the name of the object activity or an array name. Object and calling activity may be the same, i.e. argument "i" may designate the calling activity.
- j — specifies nominal length of the time interval, in units as specified by k, and as defined above. This argument shall be an integer expression.
- k — specifies the units of time as follows:
  0 - Basic counts of the system's real time clock
  1 - Milliseconds
  2 - Seconds
  3 - Minutes
  This argument shall be an integer expression.
- l — specifies a time delay, in units as specified by k, for the initial activation, as measured from the time the call was made. The time delay shall be interpreted like parameter j in the call of subroutine START. This argument shall be an integer expression.
- m — see 2.3.1.

2.3.6.2. Initial start at specified absolute time.

Execution of a reference to the subroutine CYCLAT has the same immediate effect as a reference to subroutine STRTAT. Additionally, the designated activity shall be re-scheduled, each time the activity leaves state PENDING. The cyclic re-scheduling is exactly identical to this function of subroutine CYCLE, described in section 2.3.6.1.. The interval can be modified by another call to CYCLAT, or to CYCLE, with results identical to those described in sec. 2.3.6.1.. Unpredicted delays, like those described for CYCLE, will be handled as described in sec. 2.3.6.1..
The form of the call is:

```
CALL CYCLAT(i,j,k,l,m)
```

where:

- i,j,k — are identical to the corresponding parameters of call to subroutine CYCLE (see sec. 2.3.6.1.).
- l — designates an array, whose first 7 elements contain the absolute time at which the specified activity is supposed to enter state RUNNING initially. This argument is exactly equivalent to parameter j of call for subroutine STRTAT. The elements of the array are as follows:

First element:	Year (0 →)
Second "	Month (0 to 12)
Third "	Day (0 to 31)
Fourth "	Hours (0 to 23)
Fifth "	Minutes (0 to 59)
Sixth "	Seconds (0 to 59)
Seventh "	Milliseconds (0 to 999)

  Any value outside the ranges specified shall result in an error return. Value zero for "Year", "Month", or "Day" shall be interpreted as current year, month, and/or day and should be changed to correct values automatically by the executive system (see explanation in description for CALL STRTAT). Time is interpreted as "current local time".
  This argument shall be an integer array name.
- m — see 2.3.1.

2.3.7. Connection of an activity to an event.

Execution of a reference to a subroutine CON shall cause the object activity to be associated with a specified event. This association shall cause the transition to state PENDING if, or when, the object activity is or arrives in state DORMANT. The specified event shall constitute the condition for the subsequent transfer to state RUNNING.
The form of this call is:

```
CALL CON(i,e,m)
```

where:

- i, m — see 2.3.1.
- e — specifies an integer array which contains all the information necessary to designate the event to be connected and the type of connection.

2.3.8. Elimination of an event connection.

Execution of a call to subroutine DECON shall cancel any connection between an object activity and a specified event. Thus, it eliminates further effects from a previous call to subroutine CON. If the object activity is in any executing state, the termination shall affect only future executions. Thus, the object activity will conclude its present execution, without any intervention by this call.
The form of the call is:

```
CALL DECON(i,e,m)
```

where:

i see 2.3.1.

e specifies an integer array which contains all the information necessary to designate the event which is to be disconnected.

m is set on return to the calling program, to indicate the disposition of the request as follows:

0 or less	: undefined
1	: request accepted. The object activity is neither running nor suspended.
2	: request accepted, but the activity is still running or suspended.
3 or greater	: request rejected.

This argument shall be an integer variable or integer array element, local to the calling program.

2.3.9. Elimination of previous scheduling.

Execution of a reference to subroutine CANCEL shall cancel the effects of previous calls to subroutines START, STRTAT, CYCLE, or CYCLAT for a designated, object activity.

If the object activity is in any active state (RUNNING or SUSPENDED), the elimination shall affect only future executions. Thus, the object program will conclude its present execution, without any intervention by this call.
The form of this call is:

```
CALL CANCEL(i,m)
```

where:

i see 2.3.1.

m is set on return to the calling program, to indicate the disposition of the request as follows:

0 or less	: undefined
1	: request accepted. The object activity is neither running nor suspended.
2	: request accepted, but the activity is still running or suspended.
3 or greater	: request rejected.

This argument shall be an integer variable or integer array element, local to the calling program.

2.3.10. Delaying continuation of an activity.

Execution of a reference to the subroutine WAIT shall provide a means whereby a running activity is suspended for a specified length of time. After the suspending period, the program shall resume execution, first executing the instruction immediately following the call of subroutine WAIT.

The time delay is defined as the nominal duration from the time when the call was made until the program resumes execution in its virtual processor, by being transferred to state RUNNING. The actual instants for the entering and leaving states RUNNING and SUSPENDED are subject to the resolution of the system's real-time clock and to the interrogating and activating actions performed by the Executive system. A specified zero delay shall be equivalent to no action, i.e. immediate return from subroutine WAIT. The call shall be rejected, with m=2, if a negative delay is specified.
The form of the call is:

```
CALL WAIT(j,k,m)
```

where:

j specifies the length of time delay, in units as specified by k, and as defined above. This argument shall be an integer expression.

k specifies the units of time as follows:
0 - Basic counts of the system's real time clock
1 - Milliseconds
2 - Seconds
3 - Minutes

This argument shall be an integer expression.

m see 2.3.1.

2.3.11. Synchronizing suspension of an activity.

Synchronization between two parallel activities is achieved by transfer of one activity into state SUSPENDED while waiting for certain conditions to be satisfied by the other.

Semaphores represent one of the most basic synchronization mechanisms, and this is probably also the one most widely accepted. This consideration has been dominant for the selection as the sole mechanism to be included in the proposed standard.

Semaphores are traditionally manipulated by Dijkstra's P and V primitives. [15]. Although these primitives are widely known by these terms, single letter identifiers should be avoided, since they can too easily be confused with user defined names. Also, the semaphores and semaphore operations contained here are more advanced than the common simple type, since breakpoint and increment are not confined to value 1. Thus, the semaphore manipulating subroutine calls are here described, using the suggested designations WAITS (WAIT on Semaphore), and SIGNAL, corresponding to P and V respectively.

Although it was mentioned in the introduction to this chapter, that all operations described here are indivisable, this atomic behavior must be emphasized again. Atomic, or indivisable, performance is an even more fundamentally important requirement for all synchronization operation.

The following paragraphs describe in detail the subroutine calls for the synchronizing mechanisms mentioned in the introduction above.

2.3.11.1. Wait on Semaphore.

Execution of a reference to the subroutine WAITS involves manipulation on a non-negative integer variable, referred to by one of the arguments of the call, and shared with other, parallel activities. This particular shared variable is termed "semaphore", and

the immediate effect of the subroutine call depends on the magnitude of the semaphore, as described below. The operation on the semaphore shall be exclusive, i.e. only one activity at a time may access the semaphore. Moreover, the semaphore can be accessed only by means of system calls.

The form of the call of WAITS is:

```
CALL WAITS(r,j,m)
```

where:

r — is an integer expression that specifies a reference[1] to an integer variable termed "a semaphore". The value of r uniquely specifies a semaphore that is to be shared with the parallel activities with which the calling activity is to be synchronized. Assuming s represents the internal value of the semaphore, then the effect of the subroutine call shall be:

The subroutine call is granted exclusive access to the variable s throughout its operation. If s is less than j when the subroutine is called, then further action of the subroutine call will be interrupted, the calling activity will be transferred into the state SUSPENDED and the exclusive access rights temporarily released. Since this suspension is a normal and intended operation of the synchronization mechanism under control of the operating system, it does not conflict with the basic atomic requirement mentioned before. An activity thus suspended will resume the state RUNNING and reenter at the point where it was discontinued, when the semaphore becomes greater than or equal to j again, after a CALL SIGNAL, performed by another activity. On resumption, the subroutine WAITS will perform, on behalf of the calling activity, the operation, exclusive in time:

s=s-j

and a return is then made to the calling activity. See description for CALL SIGNAL.

If $s \geq j$ when the call is acknowledged, then the calling activity is not suspended and the operation s=s-j will be performed directly, without any intervening suspension.

j — is an integer expression, specifying the amount by which the semaphore variable is to be decremented, if applicable. See description for parameter r. The value of j must be positive (one or greater), and j=1 corresponds to the simple semaphore most commonly used.

m — is set on return to the calling activity, to indicate the disposition of the request as follows:

0 or less	: undefined.
1	: request accepted, synchronization obtained.
2	: request rejected, because of non-existing semaphore.
3	: request rejected, because j is outside allowed range.
4 or greater	: request rejected, because of unspecified error.

This argument shall be an integer variable or integer array element.

2.3.11.2. Release of semaphore.

Execution of a reference to the subroutine SIGNAL shall increment an integer semaphore variable, referred to by one of the arguments of the call. Other activities, suspended in their execution of CALL WAITS relating to the same semaphore r, shall have their suspension condition re-evaluated after the present *SIGNAL* call is *terminated*. This will provide an opportunity for suspended activities to be released and resume operation, as described for CALL WAITS, parameter r above. This continuation is subject to all common restrictions pertaining to exclusive operation on a same semaphore. Thus, the effect will be, that only one of the suspended other activities will be examined at a time. After this examination, s may have been reduced again, as a consequence of releasing operation of WAITS for the examined activity. If, at this point, s is still positive, possible remaining suspended activities will be examined similarly. This examination will terminate when either no more activities are waiting for the particular semaphore, or, s becomes zero. s equal to 0 will prevent possible remaining other activities to resume the conclusion of their WAITS-call, until they be selected after another SIGNAL call. The order in which suspended activities are checked is processor-dependent.

The form of the call is:

```
CALL SIGNAL(r,j,m)
```

where:

r — is an integer expression which specifies a reference to an integer semaphore variable, shared with the activities with which the calling activity is to be synchronized. The subroutine shall be granted exclusive access to this variable during its operation and will execute the following modification of its value:

s=s+j

A change from zero to positive value, caused by this operation, shall provide a possible releasing transfer to state RUNNING for an activity waiting in state SUSPENDED for this event to occur. The semaphore value itself is not accessible directly by the subroutine call.

j — is an integer expression, specifying the amount by which the semaphore variable is to be incremented, if applicable. See description for parameter r. The value of

---

[1] This means, that r=2 specifies one semaphore, r=3 another, completely different, semaphore. In a user's program, r will usually be specified as a constant.

j must be positive (one or greater), and j=1 corresponds to the simple semaphore most commonly used.

m see 2.3.11.1.

2.3.11.3. Initialization of semaphore.

Execution of a reference to the subroutine PRESEM has two purposes:
Firstly, it introduces (declares) a particular semaphore to the system, permitting the system to give diagnostic warnings during run time, if a semaphore operation (SIGNAL or WAITS) is issued referring to a semaphore that is not declared, i.e. unknown to the system. Secondly, PRESEM establishes the initial value for the semaphore. A call of PRESEM should only be performed in the initialization phase of the execution of an RT program, and an error reaction should be issued to an activity attempting to access a semaphore that has not first been initialized. The form of the call of PRESEM is:

```
CALL PRESEM(r,s,m)
```

where:

r is an integer expression which specifies a reference to a semaphore. Usually, but not necessarily, r will be a constant.

s is the initial value given to the semaphore variable. Until the CALL PRESEM is executed for a particular semaphore, and the internal variable is assigned value s, the internal value is undefined, and another system call referring to this semaphore shall give an error return.

m is set on return to the calling activity, to indicate the disposition of the request as follows:

0 or less	: undefined
1	: request accepted, semaphore defined, value defined.
2	: request rejected, because the semaphore has already been initialized.
3 or greater	: request rejected by other reasons.

This argument shall be an integer variable or integer array element.

2.3.11.4. Reading a semaphore value.

A semaphore value may be interrogated by execution of a reference to the function RDSEM. The purpose of this function is not that it be used as a synchronization operation, but only to provide a means to supervise synchronization in a system. Thus, a reference of function RDSEM can, for example, provide information about how far a buffer or another shared resource is from saturation. When accepted (honoured), the reference will be granted exclusive access to the semaphore. If the semaphore is already being accessed by another system call when the reference of RDSEM is made, the reference will be subject to the same deferred response and contention mechanisms as the other semaphore operations. On return, the function designator will have a value equal to the internal semaphore value when the reference was accepted.
The form of the reference is:

```
RDSEM(r,m)
```

where:

r is an integer expression uniquely specifying the semaphore.

m is set on return to the calling activity, to indicate the disposition of the request as follows:

0 or less	: undefined
1	: request accepted, parameter s has been set to the value of the semaphore.
2	: request rejected, because no semaphore exists with reference r.
3 or greater	: request rejected by other reasons.

This argument shall be an integer variable or integer array element.

2.3.11.5. Waiting for an event.

A primitive but versatile synchronization mechanism is realized through call of subroutine AWAIT, with the following effect: The activity is suspended, until an event has occurred, or a maximum time has elapsed. Which condition that has caused the re-activation will be evident from the value of the *return-parameter m. The parameter j specifies the* event which subsequently will permit the calling activity to continue its execution, by being transferred to state RUNNING. The call will have no effect, if the specified condition is true already when the call is made. The form of the call is:

```
CALL AWAIT(e,j,k,m)
```

where:

e specifies an integer array which contains all the information necessary to specify the conditions for the calling activity to resume execution by being transferred to state RUNNING.

j specifies the maximum waiting time, in units as specified by k, and as defined above. This argument shall be an integer expression.

k specifies the units of time as follows:
0 - Basic counts of the system's real time clock
1 - Milliseconds
2 - Seconds
3 - Minutes
This argument shall be an integer expression.

m is set on return to the calling activity, to indicate the reason for re-activation as follows:

0 or less	: undefined.
1	: event has occurred
2	: the specified time has elapsed.
3 or greater	: request rejected, because of unspecified error.

This argument shall be an integer variable or integer array element.

2.3.12. Normal termination of execution.

Execution of the FORTRAN statement STOP shall terminate the normal execution of an activity and return the activity to state DORMANT. The form of the operation is as specified in the FORTRAN STANDARD.

2.4 Allowed and unallowed state transitions[1)]

The implementation of the subroutines defined for management of parallel activities in previous chapters requires some remarks to the semantic of these units.

The basic philosophy for the management of parallel activities is the same as in [10]:

> A certain activity is only able to exist at a certain time in one of the states shown in the state and transition diagram, Fig. 1.

Any attempt to transfer an activity into a second parallel activated state will have no success and result in the return of a failure parameter m showing a rejected request.

Example:

An activity is in state RUNNIG and attempts by a CALL CON to connect itself to an event; this call will have no success and will only result in a failure parameter m equal to 2 or greater.

This task may be handled, however, by creation of a new activity, with the same program code and by connecting this activity to the event by a CALL CON.

Multiple activations of the same activity have not been allowed in "Industrial Real-Time FORTRAN" by the following reasons:

- The RT-behavior of the system shall be simple, clear, and secure.
- The simplicity results in a simple RT Operating System which can be realized with moderate expenses.
- The cleanness results in a clear program, as no complex operations and situations are allowed.
- The user-programs can be made very secure by always testing the failure parameter after the calls; thus ambiguous situations in the management of parallel activities may be avoided.

There are further restrictions with respect to programming security, not yet discussed:

If an activity has been activated for cyclic runs by a CALL CYCLE or a CALL CYCLAT, or if an activity has been activated by a CALL CON for runs after an event, ambiguous situations might occur if these activities are transferred from state RUNNING to state SUSPENDED. In state SUSPENDED, a second activation might occur by the cycle or the event, if the activity remains too long in state SUSPENDED. Therefore, the following restrictions are defined:

(a) It is not possible to transfer an activity activated by a CALL CON from state RUNNING into state SUSPENDED.

(b) It is not possible to transfer an activity, activated by a CALL CYCLE or CYCLAT into state SUSPENDED by a CALL WAITS. This transition is possible, however, by a CALL WAIT or AWAIT, if the delay time, specified in the parameters of these calls is less than 10 % of the cycle-time.

It is, of course, possible to circumvent these restrictions. If, for example, semaphore operations are really needed in an activity connected to an event, the user may split this activity into two parts: A first activity connected to the event, and a second activity beginning after the STOP of the first activity. The second activity is activated by the first one, by a CALL START, the event disconnected by a CALL DECON, and then the activity terminated by STOP. The new activity may now use a CALL WAITS, and before termination, it must connect the event again to the first activity by a CALL CON. With this solution, one event occurring while the second activity is RUNNING or SUSPENDED is normally buffered by hardware.

The following tables summarize the allowed state transitions in "Industrial Real-Time FORTRAN". An illegal CALL will have no effect and will only return a failure parameter m, indicating that the request has been rejected.

Activation of the running activity by:	Allowed CALLs
START, STRTAT	WAIT, AWAIT, WAITS, STOP
CYCLE, CYCLAT	STOP WAIT and AWAIT only if delay time is less than 10 % of cycle time
CON	STOP

Table 1: Allowed CALLs for a state transfer of a running activity.

State of the object activity	Allowed CALLs
NON EXISTENT	CREATE
DORMANT	START, STRTAT, CYCLE, CYCLAT, CON, KILL
PENDING	KILL; CANCEL if the object activity has been activated by START, STRTAT, CYCLE, or CYCLAT; DECON if the object activity has been activated by CON.
SUSPENDED	SIGNAL if the object activity is brought into state SUSPENDED by WAITS.

Table 2: Allowed CALLs for changing the state of an object activity.

1) Section 2.4 was completely rewritten shortly before printing of the present issue (February 1978) and has not yet been discussed in the committee.

3.0 BINARY PATTERN AND BIT PROCESSING

3.1 The missing Data Types, BIT or BINARY

The arithmetic in application programs of process technology is much less characterized by the processing of INTEGER, REAL, or DOUBLE PRECISION quantities -- as, for example, in engineering-scientific programs -- than by the processing of binary patterns or single bits, which as a rule present some form of status values or packed data. Since the Industrial Real-Time FORTRAN conceived here complies with the rules of ISO FORTRAN, it is not possible to introduce new data types for the elements: binary pattern and bit. If one proceeds, however, on the premise that in a modern digital computer INTEGER values are represented by their binary values, then one can with the aid of standarized sub programs effectively process binary patterns as well as single bits. The bit number in a storage unit corresponds to the exponent of a power of two representation of positive numbers.

In the following sub programs it is assumed that INTEGER numbers are represented in binary form and negative values in two's complement form.
For example the following internal representation would be obtained with a word length of 16 bits:

Value	Binary Pattern															
Bit:	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
-10	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0

The bits are thus numbered from right to left.

3.2 Binary Pattern Processing

3.2.1 Logic Operations

Logic operations provided are Boolean functions OR, AND, EOR, and NOT. These operations are implemented as INTEGER functions. The implicit type for OR, AND and EOR is indicated by the use of I as the first letter of the function name. Their parameters, m and n, can be single variables, array elements, or expressions of type INTEGER.

After execution of the functions, the parameters remain unchanged. The operations are performed on corresponding (equal valued) bits of the two operands.

3.2.1.1 Inclusive OR

The form of this function is:
IOR (m,n)

The parameters, m and n, are combined according to the following truth table:

m	0 1 0 1
n	0 0 1 1
Function Value	0 1 1 1

3.2.1.2. Logical AND

The form of this function is:
IAND (m,n)

The parameters m and n, are combined according to the following truth table :

m	0 1 0 1
n	0 0 1 1
Function Value	0 0 0 1

3.2.1.3. Logical Complement

The form of this function is:
NOT (m)

The parameter is logically complemented according to the following truth table :

m	0 1
Function Value	1 0

3.2.1.4. Exclusive OR

The form of this function is:
IEOR (m,n)

The parameters m and n, are combined according to the following truth table :

m	0 1 0 1
n	0 0 1 1
Function Value	0 1 1 0

3.2.2. Shift Operations

The shift operations provided are logical, arithmetic and circular. The shift operations are implemented as INTEGER functions. The sub programs have two parameters, m and n.

Simple variables, array elements or INTEGER expressions are permitted.

m	specifies the value (binary pattern) to be shifted
n	specifies the shift count
n>0	indicates a left shift
n=0	indicates no shift
n<0	indicates a right shift

If the value of the shift count is greater than the number of bits in a storage unit, then the result is undefined.

The parameters are not changed by the shift operations.

3.2.2.1. Logical Shift

The form of this function is:
ISHL (m,n)

All bits representing the parameter m are shifted n places. Bits shifted out from the left end or the right end, as the case may be, are lost. Zeros are shifted in from the opposite end.

3.2.2.2. Arithmetic Shift

The form of this function is:
ISHA (m,n)

All bits representing the parameter m are shifted n places. In the case of a right shift (n<0), zeros are shifted into the left end if m is positive, and ones are shifted in if m is negative. The bits shifted out of the left end are lost. In case of a left shift (n>0), zeros are shifted into the right end while the bits shifted out of the left end are lost. In a left shift an arithmetic overflow will be indicated whenever a sign position changes.

3.2.2.3. Circular Shift

The form of this function is:
ISHC (m,n)

All bits representing the parameter m are shifted circularly n places; i.e., the bits shifted out of one end are shifted into the opposite end. No bits are lost.

3.3. Bit Processing

Individual bits of a storage unit can be tested and changed with the routines for bit processing. The sub programs have two parameters j and k. For both j and k simple variables and array elements are permitted, and k can also be represented by expressions. All parameters must be of INTEGER type.

j	specifies the binary pattern
k	specifies the selected bit

If k is negative or greater than the number of bits that are used to represent an INTEGER value, the result of the sub-program is undefined.

3.3.1. Bit Testing

The form of this function is:
BTEST (j,k)

This function is of type LOGICAL. The kth bit of parameter j is tested. If it is 1, the value of the function is .TRUE. ; if it is 0, the value of the function is .FALSE. Since the parameters are not changed by the function call, j can also be an INTEGER expression.

3.3.2. Set Bit

The form of this call is:
CALL BSET (j,k)

This operation sets the kth bit in j to a 1.

3.3.3. Clear Bit

The form of this call is:
CALL BCLR (j,k)

This operation changes the kth bit in j to 0.

3.3.4. Change Bit

The form of this call is:
CALL BCHNG (j,k)

This operation complements the kth bit in j.

4 PROCESS-INPUT/OUTPUT

The user of Industrial Real-Time FORTRAN must be able to address specific process devices for his application. As the majority of I/O systems are computer dependent, this can only be standardised in a universal way by calls to standardised driver routines which are especially written for each I/O system. It must be remembered, however, that computer independent I/O systems are either standardised, or standards are under consideration (CAMAC, GPIB, MEDIA, etc.). Standardised I/O calls designed for these systems are equally valid to the calls presented here but are outside the scope of this standard.

4.1 Scope of the Process I/O and general structure of the I/O routines

The process peripheral is the link between the process, or its terminal devices, and the central processing unit of the computer. Data describing the space and time behavior of the process are received by a processing unit and prepared so that they can be transferred to the central processing unit through an I/O interface. The variety of tasks required by the process has resulted in a large number of peripheral devices from the different manufacturers. However, in the course of many years of hardware development, largely compatible and generally accepted lines of development have become established. They may be characterized by the following statements.

Since in general, a large number of process peripherals are interfaced by uniform control units, these peripheral devices must be addressed by hardware. The multiplexing of the process peripherals is accomplished by means of input multiplexers or output multiplexers. The hardware address is used directly in the programs in order to avoid the storage and time consuming job of conversion that otherwise would be necessary.

The standard of FORTRAN specifies that one statement must be completed before the processing of the next statement begins. The standard process I/O here described corresponds to this method. The calling activity will wait for completion and this operating mode is indicated by the last letter W (waiting) of all subroutine names. [1].

The procedures for process I/O normally have four (in a special case, five) parameters which in the following will be designated in the general form with i, j, k, m (and n if needed). Such an I/O call has the general form:

```
CALL PROCEA(i, j, k, m)
```

where:

- i: Number of values to be transferred (integer).
- j: Name of an integer array that contains more complete information describing the measurement points. The orderly representation of information can not be standardized but should rather be described in the manual of the manufacturer. The array must at least contain the hardware address and data conversion information.
- k: Name of an integer array that contains the values to be transferred. In digital I/O there are as many bits to be transferred as a storage unit contains.
- m: Status indicator (integer). Its value characterizes the "success" of a call:

  0 : undefined
  1 : all data have been transferred
  2 : the transfer is not completed
  ≥3 : error conditions

According to FORTRAN conventions parameter j and k may also be array elements.

4.2 Input/Output of Analog Values

For input we distinguish between hardware implemented sequential and random input. In the first case, for sequential input, the input parameter j contains the hardware address of the first analog input; further addresses will be generated automatically. In the second case, the full sequence of hardware addresses must be given in the array j in sequential order. For output, the form is always random, i. e., all hardware addresses are given in array j.

4.2.1 Sequential analog data input

The subroutine AISQW reads a sequence of measurement values with continuous hardware addresses. The form of the call is:

```
CALL AISQW (i, j, k, m)
```

where:

- i: number of analog values to be read. The parameter is of INTEGER type.
- j: description of either hardware or software information and for the conversion of the first and the following analog values. It is the name of an integer array or of an element.
- k: array for recording the converted analog values. It is the name of an integer array or of an element.
- m: see 4.1.

4.2.2 Analog data input in random sequence

The subroutine AIRDW reads a sequence of measurement values with random hardware addresses. The form of the call is:

CALL AIRDW (i, j, k, m)

where:

- i: number of analog values to be read. The parameter is of INTEGER type.
- j: description of the hardware and software information for the conversion of each analog value. It is the name of an integer array.
- k: array for recording the converted analog values.
- m: see 4.1.

4.2.3 Analog data output

The subroutine AOW outputs a sequence of analog values from the working storage to the analog output device with random hardware addresses.
The form of the call is:

CALL AOW (i, j, k, m)

where:

- i: number of analog values (integer).
- j: either hardware or software information for the data conversion and transfer. It is the name of an integer array.
- k: array from which the analog values are output. It is the name of an integer array.
- m: see 4.1.

Other conceivable specifications of analog input/output (e. g., block mode of operation, faster/integrating mode etc.) cannot be represented by the name of the subroutine but only by a special description information in parameter j.

4.2.4 Addressing mode and scaling

The following standard is proposed for the parameters j and k for analog I/O:

The parameter j consists of a "measurement specification" for the measurement point and of a relative measurement point address. Also, the "channel device numbers" are contained in the addressing. The measurement specifications are themselves device dependent and therefore differ from user to user (8 bits per measurement point). A part of the measurement specification determines the measurement range (for example 1 V). Having k designate an array, or a single element, takes into account that, for block transfer input, a sequence of successive hardware addresses are determined by only one (or two) integer quantities and that, on the other hand, a description field is required for random input. The parameter j consists of components of an array (in some cases containing only one element), which at times must be partitioned as in the following format:

Measurement spec.	Measurement point address

For this purpose one integer quantity is normally sufficient; but in some cases one must go to two integer quantities (the higher order bit positions then contain the continuation of the measurement specification).

The parameter k (array or single element) contains the converted value. It may have completely different values depending on coding and operation mode:

- conversion mode
- number and position of decoding bits in k
- presentation of negative values
- consideration of different measurement ranges
- the computer uses a single measurement range (fixed amplifier gain)
- the computer uses several amplifier gains which, however, must be fixed for each measurement point, or
- the computer uses automatic range setting.

A general standardization such as that 2**15 means exactly 1 V is in principle not possible since the measurement ranges may vary too greatly, e. g. from 10 mV to 10 V; thus the accuracy for the representation of the coverted values could be much more inaccurate than the encoding error of the ADC. Yet the measurement range is known to the user as a part of the parameter j. Therefore the following proposal is made:

> The parameter k has to contain the converted value in such a way that 100 % of the measurement range is represented by half of the maximum value of INTEGER. This allows a measurement overrange of approximately 99 %. In the worst case of a 16 bit word for INTEGER 100 % of the measurement range would correspond to 2**14 = 16384 and the representation accuracy would be 2**(-15) or about .003 % of the measurement range.

This method has the advantage that the engineering units y can be computed from the converted values x with the same formula y=f(x) for all computers with the same integer representation without any knowledge of the special coding of the ADC used.

4.3 Digital Input/Output

For this type of input/output it is assumed that, while the effective information may be represented at times by a single bit, it will, nevertheless, be necessary to transfer whole words into or out of an integer array (for example 16 bits for each digital word).

4.3.1 Digital input

The form of the call is:

CALL DIW (i, j, k, m)

where:

i: number of digital words (integer).
j: hardware and in some case software information for conversion and transfer. It is the name of an integer array. A possible reset specification can also be contained in j.
k: array in which the digital words are stored. It is the name of an integer array.
m: see 4.1.

4.3.2 Digital output

For the output we distinguish pulse output (D igital O utput M omentary) from digital output with a permanently held value (D igital O utput L atching).

4.3.2.1 Digital pulse output

For the function DOMW the pulse duration must be indicated in a suitable form (parameter n).
The form of the call is:

CALL DOMW (i, j, k, n, m)

where:

i: number of digital values output (integer).
j: hardware information for transfer of each digital value. This parameter is the name of an integer array.
k: array representing the words to be output. It is the name of an integer array.
n: number of time units of the computer clock for the pulse duration. If the processor does not allow selection of duration, this argument is ignored but must be present. This argument shall be an integer expression.
m: see 4.1.

4.3.2.2 Latched digital output

For latched digital output (DOLW), in addition to the output field, a mask field is also required to indicate which bits are to be changed in the output. The parameter k is therefore subdivided into k1 and k2.
The form of the call is:

CALL DOLW(i, j, k1, k2, m)

where:

i: number of digital words (integer).
j: hardware information for every word that is output.
k1: array representing the digital words to be output. The parameter is the name of an integer array.
k2: designates an array whose values define digital outputs which can be changed by the subroutine. A bit set in the k2 array indicates that the digital output will be changed to the state defined by the corresponding bit position in the corresponding integer array element in k1. The order of the elements in k1 and k2 will correspond to the order in j. This argument shall be an integer array name, or an integer array element.
m: see 4.1.

5 FILE HANDLING

5.1 Introduction

In process control computer systems the case frequently arises that several parallel activities must access the same data sets stored in external storage devices. The proposed file handling scheme is intended to offer the possibility of synchronising such accesses in a simple way; i.e. it will permit at any given time only accesses which cannot disturb any parallel activity. For this purpose the two concepts of the access privilege and the access mode are introduced.

5.1.1. Access Privilege

The access privilege is intended to be a general attribute of a file. This attribute is established at the time of creation of a file; it may also be changed by a CALL RENAMW (q.v). The access privilege is binding on <u>all activities except the creating activity</u>.

The following access privileges are defined:

1. "Read/Write" permitted
2. "Write Only" permitted
3. "Read Only" permitted
4. "No Access" permitted
5. "Exclusive Access" permitted

The creating activity always has the "Read/Write" access privilege.

5.1.2 Access Mode

If an activity other than that by which a file has been created wishes to use the file, only a part of the access privilege is required in most cases. Thus the access mode has been introduced for this purpose.

The access mode is subordinate to the access privilege and must be consistent with it. The access mode is established by an activity other than the creating activity when the file is opened, and will then be recorded as the right of access for this execution of the activity. The access mode may however be changed during the execution of the activity.

The following access modes are defined:

1. "Read/Write"
2. "Write Only"
3. "Protected Read"
4. "Unprotected Read"
5. "Exclusive"

In "Protected Read" mode the activity which opens the file has read-only access to the file and is guaranteed that no other activity has or will be granted simultaneous read/write or write-only access; this ensures that the contents of the file cannot be modified during reading. In "Unprotected Read" mode, this restriction is relaxed: another activity may access the file in read/write or write-only modes, but in this case the reading and writing of the file must be correctly synchronised by the activities concerned. It should be noted that only one activity may have write access to a given file at any time; thus read/write or write-only access will be granted only if no other activity already has access in any mode other than unprotected read.

RT-FORTRAN contains calls for the following file hardling functions:

1. creation of files
2. change of file names and access privileges
3. opening of files
4. modification of access mode
5. closing of files
6. deletion of files
7. reading of files
8. writing into files

All CALLs return an error variable to the calling program to indicate correct performance and to enable synchronisation; this error variable should always be checked by the caller.

The creation of a file establishes its general and permanent attributes. These are in detail:

1. file name

   The file name is a string of text in an implementation-dependent format, containing all information necessary to identify the file uniquely. Depending on the implementation, this may include identifiers for physical unit number, volume label etc. The file name may be stored as a Hollerith constant or in an array of suitable type.

2. record length
3. maximum number of records in the file
4. access privilege

In addition to the general attributes, the creation also sets the <u>logical unit number</u> used for subsequent READ and/or WRITE in the creating activity.

If an activity other than the creating activity wishes to access a file, it must first open the file. For the current execution of the activity the opening sets the following short-term attributes:

1. logical unit number used for subsequent READ and/or WRITE by the opening activity

2. access mode

The administration of file directories will in general depend on the operating system in use. Where a free choice is available to the implementor, it is however recommended that the short-term attributes of currently open files be kept in main storage in order to minimise the number of accesses to the external storage medium.

For reading and writing of sequential files the FORTRAN statements READ and WRITE are to be used. In addition to the standard sequential files of FORTRAN, RT-FORTRAN also comprises random access files; for reading and writing these files special CALLs defined in section 5.3 are to be used.

If an activity needs no further accesses to an opened file the file should be closed to enable other activities to access the file in modes which would otherwise conflict. If a file is not closed by the program it is implicitly closed when the activity is transferred out of the running state to one of the states DORMANT, SUSPENDED, or PENDING (see chapter 2).

When a file is deleted the entire file description is forgotten and the file name thus becomes free again. The degree to which the space reserved on the external storage medium becomes available for further use depends on the file handling system implemented.

5.2 Association of Files with Activities

5.2.1 Creation of Files

The creation of random access files and sequential files within an activity is brought about by the subroutines CRFILW and CSFILW respectively. Both imply the OPEN function (5.2.3); that is, after creation is complete, the file is opened in exclusive mode for the creating activity.

```
CALL CRFILW(i,j,n1,n2,k,m)
     for random access files

CALL CSFILW(i,j,n1,n2,k,m)
     for sequential files
```

where:

- i specifies the logical unit number to be associated with the file following its creation (positive integer).
- j specifies the file name represented as a literal (Hollerith constant or array identifier).
- n1 specifies the length of each record in terms of standard integers (EQUIVALENCE may be used to relate other data types). The argument may be an expression of INTEGER type.
- n2 specifies the maximum number of records in the file. This argument may be an expression of INTEGER type.
- k specifies the access privilege to the file applying to other activities. The creating activity has all access privileges.
  - k=0 No access by other activities permitted.
  - k=1 "Read Only" - Other activities may read but not write into the file.
  - k=2 "Write Only" - Other activities may write into the file but not read it.
  - k=3 "Read/Write" - All activities may read and write into the file.
  - k=4 "Exclusive" - All activities may open the file in the exclusive mode.
- m is set on return to the calling program to indicate the disposition of the request as follows:
  - 0 or less : undefined
  - 1 : request accepted
  - 2 or greater : request rejected

5.2.2 Change of File Name

The name and access privilege of a file may be changed by the subroutine RENAMW. The file must already be open to the calling activity in exclusive mode. The ability to change the name of a file may be contingent upon other factors external to RT-FORTRAN (e.g operating system restrictions). The file remains open to the calling activity.

```
CALL RENAMW(i,j,k,m)
```

where:

- i specifies the logical unit number associated with the file by the OPEN (positive integer).
- j specifies the new file name (see section 5.2.1).
- k specifies the new access privilege to the file applying to other activities. The creating activity has all access privileges.
  - k=0 No access by other activities permitted.
  - k=1 "Read Only" - Other activities may read but not write into the file.
  - k=2 "Write Only" - Other activities may write into the file but not read it.
  - k=3 "Read/Write" - All activities may read and write into the file.
  - k=4 "Exclusive" - All activities may open the file in the exclusive mode.

m is set on return to the calling program to indicate the disposition of the request as follows:

0 or less : undefined

1 : request accepted

2 or greater : request rejected

5.2.3 Opening of File

The opening of a file which has been created either by the operating system or by another activity is accomplished by the subroutine OPEN. This operation allocates a logical unit number to the file for the calling activity and in addition establishes the access mode.

CALL OPENW(i,j,k,m)

where:

i specifies the logical unit number to be asociated with the file (positive integer).

j specifies the file name represented as a literal (Hollerith constant or array identifier).

k specifies the access mode requested by the calling activity for subsequent READ and/or WRITE. The access mode specified here must be compatible with the current access privilege (specified by the creation of the file or the last RENAMW) and with the access modes in which any other activities already have the file open.

k=1 Protected Read.

The calling activity may read the file. The request succeeds only if no other activity currently has the file open in "write only", "read/write" or "exclusive" modes. Writing by the calling activity is not permitted.

k=2 Write Only

The calling activity may write into the file. The request succeeds only if no other activity currently has the file open in "protected read", "write only", "read/write" or "exclusive" modes. Reading by the calling activity is not permitted.

k=3 Read/Write

The calling activity may read or write thé file. The request succeeds only if no other activity currently has the file open in "protected read", "write only", "read/write" or "exclusive" modes.

k=4 Exclusive

The calling activity may read or write the file; no other activity may have access. The request succeeds only if no other activity already has the file open.

k=5 Unprotected Read

The calling activity may read the file. The request succeeds even if another activity already has the file open in "write only" or "read/write" modes, provided that no activity already has it open in "exclusive" mode. Writing by the calling activity is not permitted.

m is set on return to the calling program to indicate the disposition of the request as follows:

0 or less : undefined

1 : request accepted

2 or greater : request rejected

5.2.4 Modification of Access Mode

This subroutine implicitly performs CLOSE followed by OPEN on the specified file. It permits the access mode to be modified without interruption, thus guaranteeing that no other activity can interfere by changing its own access mode between the CLOSE and the OPEN.

CALL MODAMW(i,k,m)

where:

i specifies the logical unit number currently associated with the file by the last OPEN by this activity. This argument is an integer variable or an integer array element.

k specifies the access mode requested by the calling activity for subsequent READ and/or WRITE. The access mode specified here must be compatible with the current access privilege (specified by the creation of the file or the last RENAMW) and with the access modes in which any other activities have the file open.

k=1 Protected Read

The calling activity may read the file. The request succeeds only if no other activity currently has the file open in "write only", "read/write" or "exclusive" modes. Writing by the calling activity is not permitted.

k=2 Write Only

The calling activity may write into the file. The request succeeds only if no other activity currently has the file open in "protected read", "write only", "read/write" or "exclusive" modes. Reading by the calling activity and writing by other activities are not permitted.

k=3 Read/Write

The calling activity may read or write the file. The request succeeds only if no other activity currently has the file open in

"protected read", "write only", "read/write" or "exclusive" modes. Writing by other activities is not permitted.

k=4 Exclusive

The calling activity may read or write the file; no other activity may have access. The request succeeds only if no other activity already has the file open.

k=5 Unprotected Read

The calling activity may read the file. The request succeeds even if another activity already has the file open in "write only" or "read/write" modes, provided that no activity already has it open in "exclusive" mode. Writing by the calling activity is not permitted.

m is set on return to the calling program to indicate the disposition of the request as follows:

0 or less : undefined

1 : request accepted

2 or greater : request rejected

5.2.5 Closing of File

The closing of a file is accomplished by the subroutine CLOSEW, which cancels the access mode of the calling activity and deallocates the logical unit number.

CALL CLOSEW(i,m)

where:

i specifies the logical unit number currently associated with the file to be closed (positive integer).

m is set on return to the calling program to indicate the disposition of the request as follows:

0 or less : undefined

1 : request accepted

2 or greater : request rejected

5.2.6 Deletion of File

The subroutine DFILW permits an activity to delete a file from the file system. The file must already be open to the calling activity in exclusive mode. This call also deallocates the logical unit number associated with the file.

CALL DFILW(i,m)

where:

i specifies the logical unit number currently associated with the file (positive integer).

m is set on return to the calling program to indicate the disposition of the request as follows:

0 or less : undefined

1 : request accepted

2 or greater : request rejected

5.3 Data Transfer

This section contains proposals for subroutine calls for reading and writing of random access files. It is considered important that these subroutines should be implementable using the extended READ/WRITE calls in the proposed ANS FORTRAN of [12].

The maximum number of data transferred by the following CALLs RDRW and WRTRW are the data of one record. When the record is longer than the length specified by parameter n5 (see below), the remainder is discarded in reading and set to zero in writing.

5.3.1 Read Random Access File

CALL RDRW(i,n3,n4,n5,m)

where:

i specifies the logical unit number currently associated with the file to be read (positive integer).

n3 specifies the record to be read from the file (positive integer).

n4 indicates the first variable of a sequence to be read in. This should be the name of an array of appropriate type.

n5 specifies the length of the area to be transferred in terms of standard integers (EQUIVALENCE may be used to relate other data types). The argument may be an expression of INTEGER type.

m is set on return to the calling program to indicate the disposition of the request as follows:

0 or less : undefined

1 : request accepted

2 or greater : request rejected

5.3.2 Write Random Access File

The corresponding WRITE call is:

CALL WRTRW(i,n3,n4,n5,m)

where the arguments are to be intepreted in the same way as for RDRW.

5.4 Example of File Access by Activities

5.4.1 Creation of a Random Access File

The file is required to have the following attributes:

Logical unit number to be used for data transfers: 3
File name: MST15
Number of integers per record: 80
Maximum number of records: 200
Access privilege:
All activities may read and write the file. Special restrictions are established only by opening it in other activities.

```
CALL CRFILW(3,5HMST15,80,200,3,MF)
```

After the call, the value of the variable MF must be checked as an error parameter. In the case of success, the file is automatically open for use by the creating activity and may be processed by READ and WRITE using the logical unit number 3.

When reading and/or writing are finished, the creating activity should close the file with

```
CALL CLOSEW(3,MF)
```

After this call, the value of the variable MF is checked for errors.

5.4.2 Opening a File

Another activity processes the data of the file created above in 5.4.1. Here, logical unit number 21 is assigned to the file and the file is only read:

```
CALL OPENW(21,5HMST15,1,MF1)
```

After the call the value of the variable MF1 is checked for errors. If the call was successful, further calls may refer to the file by its logical file number 21.

5.4.3 Modification of Access Mode

In the same activity as above in 5.4.2, data are to be written back into the file after processing. For this purpose the access mode is changed to "Write Only".

```
CALL MODAMW(21,2,MF2)
```

After the call, the value of the parameter MF2 is checked for errors.

5.4.4 Closing a File

After writing back the processed information the file is closed by this call:

```
CALL CLOSE(21,MF3)
```

After the call the value of the variable MF3 is checked for errors.

5.4.5 Deleting a File

In a third activity the file MST15 is to be deleted from the file system, so it must first be opened in exclusive mode.

```
CALL OPENW(1,5HMST15,4,MF4)

CALL DFILW(1,MF4)
```

After each call, the value of the variable MF4 is checked for errors.

5.4.6 Renaming a File

An activity wishes to rename a file from RUN27 to DATA, and at the same time change the access privilege for other activities from "Read/Write" to "Read only". For this operation, the file must first be opened in the exclusive mode, then renamed and immediately closed.

```
CALL OPENW(1,5HRUN27,4,MF5)

CALL RENAMW(1,4HDATA,1,MF5)

CALL CLOSEW(1,MF5)
```

The error parameter is checked after each call.

References

[1] ISO Recommendation R 1539, Programming Language FORTRAN. ISO, Geneva (Switzerland), 1972.

[2] Hohmeyer, R. E.: CDC 1700 FORTRAN for Process Control. IEEE Transactions on IECI, Vol. 15, No. 2, Dec. 1968.

[3] INDAC 8. Digital Equipment Corporation, 1969.

[4] DDP-516 OLERT, Programmer's Reference Manual. Honeywell Inc., Doc. No. 1300 72069 A, EM-602, 1969.

[5] Bonnard, P.: PROCOL: A Programming System Adapted to Process Control. IFAC Congress, Paris, 1972.

[6] ISA S61.1 (1972) Standard. Industrial Computer System FORTRAN Procedures for Ececutive Functions and Process Input-Output. Instrument Society of America, 1972.

[7] ISA S61.2 (1973) Draft Standard: Industrial Computer System FORTRAN Procedures for Handling Random Unformatted Files, Bit Manipulation, and Date and Time Information. Instrument Society of America, 1973.

[8] ISA S61.1 (1976) Standard. Industrial Computer System FORTRAN Procedures for Ececutive Functions, Process Input-Output, and Bit Manipulation. Instrument Society of America, 1976.

[9] ISA S61.2 (1977) Draft No. 2 Standard: Industrial Computer System FORTRAN Procedures for File Access and the Control of File Contention. Instrument Society of America, 1977.

[10] G. Heller, H. Mittendorf, G. Teuschler, M. Topschowsky, C. Wustlich: Prozess-FORTRAN 75, eine Erweiterung von FORTRAN für Prozess-rechner-Anwendungen. VDI/VDE-GMR, 1975.

[11] O. Pettersen: Management of Parallel Activities in Real-Time FORTRAN. State Model and State Transitions. 2nd edition. Minutes of the Purdue Europe Spring Meeting 1977 at ISPRA CCR Euratom, Vol. II. IRIA, Rocquencourt, 1977.

[12] Draft Proposed ANS FORTRAN. BSR X3.9, X3J3/76. SIGPLAN Notices, Vol. 11, No. 3, March 1976. Association for Computing Machinery, New York, 1976.

[13] P. Brinch Hansen: Operating System Principles. Prentice-Hall, 1973.

[14] A. C. Shaw: The Logical Design of Operating Systems. Prentice-Hall, 1974.

[15] E. W. Dijkstra: Cooperating Sequential Processes. In "Programming Languages" (V. Genuys, ed.), pp 43 - 112. American Press, New York, 1968.

APPENDICES E-VIII

TC-3

LONG TERM PROCEDURAL LANGUAGES COMMITTEE
PURDUE EUROPE

1. Minutes, 39th LTPL-E Meeting, October 17-19, 1977, Brussels.

2. Minutes, 40th LTPL-E Meeting, January 25-27, 1978, Brussels.

3. Chalmers, A. F., Kronental, M., and Shorter, D. N., _Proposals for the Way Ahead for LTPL-E_, April 4, 1978.

Author J.W. Roberts

Institution

mbp GmbH
Waldweg 7
D2120 Lüneburg

LTPL-E/JR 780303

Category M

Replaces:
LTPL-E/JR 771026

Title

Minutes of the 39th LTPL-E-Meeting Brussels,
Oct 17th to Oct 19th 1977.

PRECEDING PAGE BLANK-NOT FILMED

Present:

J.G.P. Barnes	ICI Ltd
A.F. Chalmers	GEC Computers Ltd
R.M. De Morgan	DATASKIL Ltd
P. Elzer (Chairman)	Uni Erlangen
T.J. Froggatt	Uni York
A.L. Golborn	Systems Designers Ltd
J. Heger	BBC
M. Inderst	ESG
R.F. Maddock	IBM UK
J. Robert	CAP
J.W. Roberts	MBP GmbH
S. Savoysky	LCPC
C.H. Smedema	Philips
I. Smith	Digital Equipment Co.
K.H. Timmesfeld	IDAS GmbH
G. Verroust	IPN/Université Paris-Sud
H.B. Williams	MBP GmbH

Apologies for absence from:

G. Bianchi	DPHN/HE
R. Gilbert	AERE
J.D. Ichbiah	CII
A. Kappatsch	IDAS
M. Kronental	IRIA/CTI
J. Levy	IRIA/STI
I.C. Pyle	Uni York
W.E. Quillin	Plessey
E. Wegner	GMD

Technical Session 19th Oct 77, 9.00

T1 Approval of agenda

The agenda was accepted

T2 Report on full meeting of IPW at Purdue, by Mr. Elzer.
Mr. Elzer had already circulated a report (LTPL-E/PE771014). In addition he stated that the meeting had been held earlier than usual (5th - 6th Oct) to enable European academics to attend; however few Europeans were present. Total attendance was about 80.
In general the discussions could be described as 'civilised'; there was nothing dramatic. The meeting structure had been reorganised to allow more tutorials. One was by Mr. Gaspar about his experiences with the use of micros in the chemical industry. He criticised the careless use of micros. Micros would not eliminate the need for higher level languages in process control. The motion discussed at the last Purdue-Europe meeting at Ispra about cooperation between Purdue Commitees and the DoD-HOL effort was approved with the deletion of the word 'very' in the expression 'work is very similar'.

Mr. Reh has resigned as chairman of LTPL-C. Mr. Elzer did not wish to take over because of pressure of work at the moment. Mr. Adams agreed to take the job but did not attend most of the LTPL-C meeting at Purdue, which was therefore chaired by Mr. Elzer.

The LTPL-C sessions were attended by about 9 people including two from DoD but no one from LTPL-J. Mr. Elzer gave a report on LTPL-E work (see PE771014).
Mr. Gertler gave a presentation on his work on real-time extensions to APL. Mr. Schwarm (a new member of LTPL-A) gave a presentation on PASCAL, expressing the opinion that PASCAL is cheap to implement and safe to use. PASCAL is of importance to LTPL-C because all four DoD contracts are taking PASCAL as a basis for language design. An afternoon was devoted to a

discussion on the coming PLIP meeting in London in November. American participants of PLIP expressed the opinion that the work done by the DIN group for PLIP was poor. LTPL-C decided to try to synthesise IRONMAN with LTPL-C functional requirements. A list of relevant papers was made and ordered according to chapter of IRONMAN.
The work of synthesis was divided among the LTPL-C groups LTPL-A, LTPL-E algorithmic subgroup and LTPL-E evaluation criteria subgroup. (See PE771014). It was noted that IRONMAN has no chapter on environment description.

Mr. Whitaker had reported to LTPL-C on the progress of the DoD project. 18 bids were recieved for phase 1 and 4 contracts have been awarded (see Appendix 1 to these minutes).
The contracts run to the end of February 78. A three-week evaluation will occur in March or April. LTPL-E has been invited to participate. DoD is also now working on benchmark programs and software tools; a paper comparable to STRAWMAN on software tools will be produced by 1st Dec. Mr. Whitaker is responsible for economic analyses; first results of these are expected of the end of October. They are expected to produce the conclusion that the high-level-language project is worthwhile. Phase 2 of the DoD-HOL project will be launched in mid-78; probably there will be two contracts. Final language selection occurs in April 79 and the language shall become available in 80.

Discussion

Mr. Barnes and Mr. Chalmers felt that the 'invitation to participate in evaluation' is too vague. Mr. Barnes has heard from Mr. Fisher that the DoD is hoping for unbiased opinions from European experts, whereas the qualified people in the states are mostly already in some way involved with the project.

Mr. Smedema asked whether, failing guidelines from DoD, we have our own proposals how we should participate. Mr. Elzer said we could have no firm plans until we know what support we will get from the EEC.

T3 <u>Algorithmic Subgroup Report to Plenary</u> by Mr. Chalmers

The subgroup was attended by only five people. Concern was expressed that a number of our long-established members seem no longer able to attend, resulting in some lack of continuity in our work.

We started with review of the minutes of our last meeting. The main actions had been completed including issue to the A-list of the Algorithmic proposals which give the state of the art up to end March (This is document reference RG/770412). The next draft of this document, updating it to June, was reviewed.

The discussion on arrays held in June was resumed and agreement was reached on some points. This ended the Monday afternoon session.

The Tuesday session started with a summary by the subgroup chairman of the outcome of discussions held at the Monday evening meeting of the Planning Group. In consequence it was decided to start work on a document which would round off the Algorithmic Subgroup work carried on since its inception nearly three years ago. The overall structure of this document was agreed upon. In particular the principal section of the document is to be derived from the Algorithmic proposals giving the state of the art up to June, and the rest of the session was devoted to a first draft of this section . It is intended to use the extended meeting of the Algorithmic subgroup on Thursday and Friday morning of this week to continue this work (However only three members are able to attend this extended session).

Following this, the chairman will circulate a completed draft to all members of the subgroup for comment with the intention of issuing an agreed document to the A-list before the end of the year. This document giving Current Algorithmic proposals should serve as a companion document to the Current Tasking proposals produced by the Tasking Subgroup at their special meeting in September.

The Algorithmic Subgroup will try to turn its attention after this to their part of processing Functional Requirements in accordance with the work plans agreed by LTPL-C at the IPW Fall Meeting in October. To do this however, it is necessary to optain copies of the numerous documents identified as relevant by LTPL-C.

Discussion

None

T4 Report on Evaluation Criteria Subgroup

Mr. Elzer: Mr. Levy has resigned as chairman. The only member present was Mr. De Morgan.

Mr. De Morgan: Mr. Levy has missed the last three meetings. This time he even sent no representative. I didn't know he was going to resign.

Mr. Elzer: You have produced a paper. Is it being distributed.

Mr. De Morgan: It is based on an unrevised IRONMAN, so it needs updating. It's not a very exciting paper, the main conclusion is that IRONMAN is reasonably consistent on the whole.

T5 Report on I/O Subgroup by Mr. Verroust

We met fruitfully with 4 members.
We received 1 written and 2 oral apologies E. Wegner,

A. Kappatsch, Mrs. G. Bianchi.

We pursued the work defined in January and begun at last meeting.

1. Revision of some points of the last version of our technical proposals report LTPL-E/335. These revisious led to add some explanations to some unclear or ambignous points.

2. Addition of a new part concerning the elementary I/O operations for transferring data and status or control information to/from our file.
   We defined 4 elementary operations and a type of data named STATUS.
   A status is a data object containing information about the internal state of the file from the channel to the unit controlling/reading the physical parameter.
   We defined 2 parts in a status.
   a -a permanent standard part with data defined with language definition
      ex.: - disconnected
           - usable to operate
           - busy

   b -an extensible part defined by the programmer and containing special status data which the physical interface is able to extract from the control/reading unit.

   - the permanent part can be defined by a declaration or implicitely by an I/O statement

   - the extensible device-dependant user part must be defined by an explicit declaration.

   The new version of LTPL-E/335 will incorporate these new proposals.

3. We discussed some other problems but without any definitive proposal:

   - formatting language
   - general file statements: connexion
     general status
   - external definition of a file and sharing of a file by several tasks.

   For this last point, we'll examine the last proposals of tasking group to see if these proposals fit with our needs.

4. We had a discussion on the problem of the disappearing of explicit connected happening in tasking proposals.
   In an I/O-operation, the connected happening has a physical sense, even it can appear in different ways at high level.
   The loose happening can be defined as a special case of connected happening with a special status, but the connected happening is a necessity.
   We would have a discussion with tasking subgroup after analyses of their last report.

5. On Tuesday afternoon, we had a discussion on the consequences of the possible official EEC decisions on the work of our subgroup.
   Our work has some general scientific interest in the state of art and is almost independent from the existence of LTPL-project. We are worried about some way to continue our work.

6. Our next meeting, if it exists, will be devoted to a first definition of general file handling operations based if possible on an automaton state model of our file.

- Received papers:

  A. KAPPATSCH's comments on LTPL-E/335-1
  was discussed and led to some incorporations

  P. ELZER LTPL-E/PE770929
  "Homogeneous addressing, an alternative view of I/O in distributed systems"
  was briefly discussed and considered as out of the scope of our immediate work.

Discussion

Mr. Timmesfeld: The meaning behind the words 'connected happening' has not been abolished, merely the words no longer appear in the description of synchronisation.

Mr. Elzer: Will the I/O Subgroup's latest working paper be distributed.

Mr. Verroust: Yes.

T6 Report on Tasking Subgroup by Mr. Timmesfeld

Our start was delayed until Tuesday morning because several members were held up by fog. Our main work was to review the paper TG770923 produced by Kronental, Roberts, Timmesfeld and Wand in the week of September 19th to 23rd. The paper was slightly modified for increased clarity and to remove typing errors. The approved version will be circulated. We then discussed the future of the tasking subgroup and concluded

that since there is now no great interest in the definition of a real-time process control language for Europe, the tasking subgroup should stop its work leaving TG770923 as its final document.

Discussion

Mr. Inderst: Explain how connected happenings are now handled.

Mr. Timmesfeld: See the remark at the end of the introduction to section 3, Happenings. What were formaly refered to as connected happenings are explained in section 5, Synchronisation, without the necessity for the words 'connected happening' to occur.

Mr. Barnes then suggested that TG770923 be published in the technical literature. There was discussion of the merits of publishing the final statements of all our subgroups.
The meeting concluded that the authors should decide whether they wished to have the paper published, but if so it should contain an introduction relating it to the LTPL-E effort.

T7 Other technical matters

None

Business Session

B1 Approval of previous minutes

Mr. Elzer and Mr. Timmesfeld offered minor corrections to the minutes.
The letter from Mr. Elzer to Mr. Reh about the Ispra motion on cooperation with DoD, which should have formed Annex 2

to the minutes, was never written. The matter was settled by phone calls and discussion in the LTPL-C meeting at Purdue, as reported this meeting under T2.

B2 Report of planning group and discussion

plus B3 Information on US-DoD-HOL project

plus B4 Information on European project

These items could conveniently be handled together.
Mr. Elzer reported.
The planning group held two meetings, one on Monday and one on Tuesday, each more than two hours.
The first meeting was needed as preparation for the meeting of officials which took place Tuesday, the main item of whose agenda being the LTPL-E project and to which I gave a presentation. The questions before the planning group were, what can we do in the long term and what must we do in the short term. In the short term we will act as a forum for people who wish to monitor and influence the DoDHOL project.

The planning group came to no decision on long term activities. Two possibilities were discussed,

1. the European group should try to define a language smaller than the DoDHOL.

2. It could do something genuinely European, in a field where we are not yet behind the Americans, namely design a very high level language, working on the lines of MASCOT/MORAL and similar work which we know of in the USA and the Federal Republic of Germany.

The staffing problems of the I/O and Evaluation Criteria subgroups were discussed. One proposal was that the tasking and Algorithmic subgroups finalise their work and interested members join one of the other groups.

Mr. Thompson of the EEC who has taken over Mr. Diettrich's position had explained to the planning group the background to the meeting of officials and invited Mr. Elzer to give a presentation on DoD and LTPL-E work.
The planning group requests position papers on the subject of future work.

In the second planning group meeting (Tuesday evening), Mr. Thompson reported on the meeting of officials. Mr. Thompson, Mr. Elzer and Mr. Garric had presented views on future work. The main discussion had been on a telex from the German delegation proposing expansion of item 4 of the project plan. No agreement had been reached, partly because of misunderstandings of some of the terms used, partly because some delegates really want nothing to be done. The planning group paper from June which proposes support for seven items had been discussed. There had been agreement that cooperation with DoD was useful and that these seven items could form a foundation for monitoring DoD work, although the British delegate had strongly questioned one of the items. Mr. Thompson had been requested to draft a plan by the end of the week showing how monitoring of and cooperation with DoD should be carried out, what it would cost etc.
Mr. Elzer and Mr. Thompson had got together about this after the meeting of officials. They then suggested to the planning group that LTPL-E has two 'time windows' in which DoD can be influenced, the first in March-April 78 and the second in 79 when the final language selection is made. Nothing can anyway be done before March 78; the EEC takes a long time to approve even a small project. Mr. Thompson and Mr. Elzer propose to find people with knowledge of the relevant fields in process control,

if possible from the ranks of LTPL-E, these people to visit the States in March to learn the results of phase 1 then after return to gather the reaction of European industry to the phase 1 proposals by organising a conference and presenting them. After this there would be some technical support studies to summarise the European industrial control requirements.
This would be support for our work but not the project we previously wanted. It would still be a lot of money. Mr. Thomson will propose this at the end of this week.

Mr. Elzer had reported to the planning group about a phone call from the publication 'Computing' asking why the project was stuck. He had refused to reply without consulting the group. However 'Computing' had published an article.

The question of publication of final papers of subgroups, in particular TG 770923 'Current Tasking Proposals' had also been raised in the planning group but without any decision.

The planning group repeats its call for manpower for the I/O subgroup. I/O is very important in the industrial environment and IRONMAN contains little on it.

Mr. Elzer also reported on an invitation from the Danish Computer Society to visit Denmark in December 77 and present the plans of LTPL-C and LTPL-E. Originally he had been willing to do this but now doubted whether he sould make such a presentation so early. He has therefore proposed February 78, by which time the results of the latest effort to get European agreement should be known.

Discussion:

Mr. Robert: When was the previous project plan rejected?

Mr. Elzer: Never. But it was also not accepted by the officials as a basis for a project.

Mr. Robert: Then the project plan is valid but delayed. It DoDHOL is found not suitable for industry, the project plan could made effective again.

Mr. Elzer: Yes, that is our official view.

Mr. Robert: There are now two project plans, the old one and the new one proposing monitoring of DoDHOL.

Mr. Elzer: Monitoring of the DoD project is an item of the old plan which has now been emphasised. The plan of autumn 76 is still our proposed project plan; the paper of June 77 has the nature of an aide-memoire for Mr. Layton.

Mr. Chalmers: Can we plan the reorganisation of groups? The tasking group has terminated itself and the algorithmic group will produce its final paper by December, although it now has a big job from LTPL-C (See T2 above and Mr. Elzer's paper PE771014) with a deadline of March. The evaluation criteria group has shrunk to one man although it is the most important group for cooperation with DoD.

Mr. Robert: Now that the project plan is dormant I expect many people will drop out for a while; we may be left with perhaps 16 apostles. With such a small group we can work without subgroups. We should return to the structure of 5 years ago, when everybody worked on every problem.

Mr. Elzer: I agree in principle. It would be good for everybody to get up to date with the work of the subgroups. However I don't think 16 people can work

efficiently. We could adapt your proposal to the planning group for formation of 'task forces' who would meet between full meetings to write position papers.

Mr. Robert: There is no more need for subgroups with little contact.

Mr. Elzer: That means discussion in series rather than in parallel.

Mr. Robert: Happily the need for synthesis is arising at just the time that the group is shrinking to a size where it could do it.

Mr. Chalmers: The algorithmic group has suffered from shifting membership. I think the I/O subgroup should continue as a separate subgroup.

Mr. Elzer: I propose as a compromise that only two subgroup meetings be held at the January meeting, namely I/O and algorithmic with about 8 attendees each. Perhaps we can make further structural changes for March.

Mr. Robert: Each subgroup should prepare a workplan and invite the whole group to help in the work.

Mr. Elzer: I will order sticky labels for Mr. Chalmers and Mr. Verroust so that they can each invite the whole group.

Mr. Barnes: You reported a proposal from you and Mr. Thomson that representatives of industry go to the States in March and report to a conference in Europe thereafter. Is this in the hope of influencing DoD through the second 'time window'.

Mr. Elzer: Yes

Mr. Robert: The idea is to keep abreast of developments.

Mr. Elzer: Under B3, information on DoD, I can read you a paper I recieved on 10th Oct.
(The paper is included as Appendix 1).

B5 <u>Discussion of contacts with ISO/TC97/SC5/WG1 - 'PLIP'</u>

Mr. Elzer: There is a PLIP meeting from 7th to 10th November in London. I won't be there.

Mr. Robert: We are failing in our attempts to influence PLIP because we didn't make the proper structure. PLIP has one meeting per year and tasks are defined by written resolution. We can best adapt to this by reporting on PLIP contacts not before but after a PLIP meeting. We can then look at PLIP's workplan for the year and see if any item is important to us, then we can establish a task force to start work on this item and so enter to PLIP cycle. If this meeting were to propose anything which I then presented to PLIP in November, it could not be accepted at that meeting but only in the following year.

Mr. Elzer: I will put PLIP on the agenda for January.

It was established that three of the LTPL-E members present would be attending PLIP, so reports would be forthcoming.

B6 Meeting structure, mailing list, paper list, etc.

Meeting structure was discussed under B2 - 4. The January meeting will be structured like this one but with only two subgroups.
The mailing list will be purged as per previous policy; all who have missed the last two meetings without sending appology and are not on the list ex-officio will be removed.

The only objection to the new paper numbering scheme had been the difficulty of knowing whether one had a full set. To overcome this it was agreed to ask the librarian Mr. Ford to circulate the paper list four times a year. Mr. Chalmers will write to Mr. Ford about this.

B7 Next meetings

The next meeting is Jan 25 - 27 in Brussels, as already agreed. The following meeting is at the Spring Purdue Europe, which will be April 4 - 7 in Zürich without EEC funding. The Purdue Europe agenda is included as Appendix 2 to these minutes.

B8 AOB

Mr. Elzer had information on 4 conferences.

1. STATE of THE ART and Future Trends in Compilation,
   Montpellier Jan 9th - 20th, 78
   Université des Sciences et Techniques du Languedoc,
   Avenue d'Occitanie
   34075 MONTPELLIER
   France

2. Sûreté de Fonctionnement
   IRIA, 28 Nov - 2 Dec 77

3. Programming Foundations
   Toulouse 5 - 16 Dec 77
   Université Paul Sabatier
   118 route de Narbonne
   31000 Toulouse
   France

4. Global Descripture methods for syncronisation in realtime applications
   3 - 4 Nov 77
   IUT Paris V
   143 Avenue de Versailles
   75016 Paris

End of meeting

Appendices

1. Information on DoDHOL
   (Refered to in T2 and B3)

2. Agenda for Purdue Europe 78
   (Refered to in B7).

APPENDIX 1 TO MINUTES OF LTPL-E 39TH MEETING

DEFENSE ADVANCED RESEARCH PROJECTS AGENCY
1400 WILSON BOULEVARD
ARLINGTON, VIRGINIA 22209

The Defense Supply Service Washington has announced the award of four contracts to produce competitive prototypes of a common high order computer programming language for Department of Defense embedded computer systems. These awards came as a result of a request for proposal and offers received from fourteen firms, both U.S. and foreign. The successful contractors were Honeywell (CII-Honeywell Bull), Intermetrics, Softech, and SRI-International.

While different approaches were offered, all four winning contractors proposed to start from the computer language PASCAL as a base. They will provide modifications to construct a resulting language to satisfy military needs as expressed in the "DoD Requirements for High Order Computer Programming Languages (Revised IRONMAN, July 1977)".

The contracts provide for three phases at the discretion of the government. The first phase is to be six-months and will produce a preliminary language design. At the end of the first phase, an evaluation of the products will result in some of the contractors being continued through full formal design, rigorous definition, and prototype implementation. The one contractor whose language is selected by the government will be continued for refinement and initial maintenance. The language will be ready for initial use in 1979.

This language design is the next step in a Department of Defense effort to reduce software costs of embedded computer systems. Earlier actions included issuing DoD Directive 5000.29, "Management of Computer Resources in Major Defense Systems," which, as one of several management actions, required the uses of approved high order languages in future Defense systems software. DoD Instruction 5000.31,"Interim List of DoD Approved High Order Programming Languages," stopped proliferation by approving only seven existing languages.

APPENDIX 1 TO MINUTES OF LTPL-E 39TH MEETING (Cont.)

The technical effort in high order languages has, over the last three years, brought increasingly refined sets of requirements, produced an evaluation of existing languages, and has established the technical feasibility of a single language for these applications. The successful design of such a language will be followed by testing and evaluation, compiler and tool generation, and the necessary long-term language control. This program is presently being directed by the DoD High Order Language Working Group, chaired by Lt. Col. William A. Whitaker, Defense Advanced Research Projects Agency.

EIDGENÖSSISCHE TECHNISCHE HOCHSCHULE ZÜRICH
ÉCOLE POLYTECHNIQUE FÉDÉRALE ZURICH
POLITECNICO FEDERALE SVIZZERO ZURIGO

Hybrid-Rechenzentrum CH-8044 Zürich, Voltastrasse 18 Telephon (01) 32 62 11, intern 2830

October 13, 1977

Dr. Th. Lalive d'Epinay

To all

PURDUE EUROPE chairmen

Regional Meeting of PURDUE EUROPE April 4 - 7, 1978 in Zürich

As you probably will know the next Regional Meeting of PURDUE EUROPE will be organized in Zürich by the ETH with the following preliminary agenda:

April 3	1500	:	meeting of the PE- and TC 1-8/E chairmen
4	0830	:	registration
	0915	:	introduction by PE-chairman
	0930	:	overview over TC 1-8/E work
	1030	:	coffee
	1100-close	:	TC meetings
	1230	:	lunch
5	0900	:	tutorial on Standardization Methods (N. Malagardis)
	1000	:	coffee
	1030-1600	:	TC meetings
	1230	:	lunch
	1600	:	round-table
6	0900	:	tutorial on Operating System Functions (TC-8 Report, Th. Lalive d'Epinay et al.)
	1000	:	coffee
	1030	:	round-table on Operating System Functions
7	0900-1100	:	TC meetings
	1100-1200	:	closing session
	afternoon	:	TC meetings according to announcement by TC-chairmen

To all PURDUE EUROPE chairmen

We intend to offer lunch and coffee in a newly build nice Mensa. The total price for the meeting to cover lunch, etc. and printing of minutes will be 80 - 100 Swiss francs.

Please send to me, as soon as possible, the following informations:

- comments/requests/etc. to the agenda
- general suggestions/hints for the meeting
- number of required invitations you need for distribution (in your TC as well as other possibly interested persons).

Thank your very much for your cooperation.

Yours sincerely

Th. Lalive d'Epinay

LTPL - EUROPEAN GROUP

Author: R. Gilbert	LTPL-E/ RG780308
Institution: Building 7.12, AERE, Harwell, Oxfordshire, OX11 ORA, England.	**Category:** M **Updates:** None **Replaces:** None **Status:**
Date (assigned): 25th January 1978	**Distribution:** A list
Date (completed): 8th March 1978	**pp:** 13

Title: Minutes of the 40th LTPL-E Meeting, Brussels, 25th-27th January, 1978.

Contents:

1. Present

Verroust, G.		IPN/Université Paris-Sud
Savoysky, S.		LCPC
Robert, J.		CAP-Sogeti
Chalmers, A.		GEC Computers Ltd.
Shorter, D.N.		BSC
Holmes, G.W.		Systems Designers Ltd.
Gilbert, R.	(Secretary)	AERE, Harwell
Wegmann, P.		ETH, Zurich
Somers, P.	(Part time)	EBES
Williams, H.B.		MBP GmbH
Smith, I.		Digital Equipment
Froggatt, T.J.		University of York
Elzer, P.	(Chairman)	University of Erlangen
Kronental, M.		BNI/IRIA
Thompson, K.	(Part time)	CEC
Gohner, P.		IRP Universität Stuttgart
Kappatsch, A.		IDAS
Helfert, M.E.		GADV
Malagardis, N.	(Part time)	IRIA

2. Apologies for absence

de Morgan, R.M.	Dataskil
Roberts, J.W.	MBP GmbH
Heger, J.	BBC
Pyle, I.C.	University of York
Wand, I.C.	University of York
Maddock, R.F.	IBM UK
Quillin, W.E.	Plessey
Teller, J.	Siemens AG
Barnes, J.G.P.	ICI

3. Discussion of connections with US-DoD-HOL project (Agenda item B4)

ELZER: We have had some bad news from the CEC: Mr. Leyton has given up trying to get an LTPL project under way even as far as funding people to go to the States to take part in the review of phase I of the DoD HOL project.

On the 16th February, a set of documents will be made available to those people who have agreed to take part in the phase I evaluation. The results of the evaluation must be in Mr. Whitaker's office by 13th March.

SHORTER: Nick Neve has sent a telex to Mr. Whitaker expressing LTPL-E's interest (Appendix 1). The Ministry of Defence has offered to pick up the proposal documents from the States and deliver them to the UK.

CHALMERS: A second telex was sent (Appendix 2) stating that a decision about LTPL-E's participation is to be taken at this meeting.

ELZER: I originally planned a meeting on the 20th February for evaluating the phase I proposals. We should decide who is going to the States from March 13th to 27th to review the evaluations.

HOLMES: There are 200 evaluation centres, they're not all going to send people to DARPA. Why should LTPL?

ELZER: I just want to discuss possibilities. We should ask Mr. Whitaker if we can send someone.

In the meantime we must decide how to do the evaluation. One possibility is that we have two meetings

. 20th February - 21st February to collect and study the material

. 6th March - 8th March to consolidate the evaluation.

HOLMES: A UK member could bring the papers to the first meeting.

Who will comprise the evaluation team?

CHALMERS: I have a short list of UK people who are willing to participate.

Ian Wand - only definite for a meeting on the 27th February

Bob Maddock - O.K. for the weeks starting 20th or 27th February.

There are other members in the UK, myself included, who are already involved in the two UK evaluation groups (one Civil group and one Military group).

ROBERT: There's a simple problem here. Our companies need to be refunded above the level of personal expenses; 3 weeks is a long time to spend away from work.

ELZER: It is still possible to arouse interest. There are national evaluation teams. What is happening in France?

KRONENTAL: CPM, Thompson and possibly CAP-Sogeti will be taking part but they will not be setting up teams; there will just be three individuals. (M. Parayre for CPM and M. Ruggiu for Thompson).

ROBERT: My company has asked that I be involved in the LTPL evaluation.

ELZER: Which bodies are organising the British teams?

SHORTER: The Ministry of Defence has its own team and the Department of Industry is organising the other team which will consist of myself, A. Chalmers, K. Clements, J. Barnes and one other.

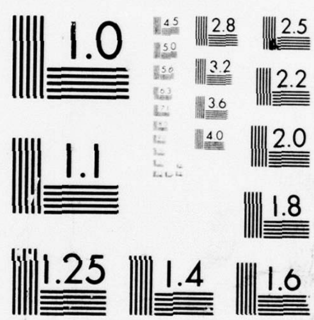
KRONENTAL: In France there is no co-ordinated team. I had hoped that the LTPL group would organise one.

AD-A068 266 PURDUE UNIV LAFAYETTE IND PURDUE LAB FOR APPLIED IND--ETC F/G 9/2
MINUTES 1978 SPRING REGIONAL MEETING INTERNATIONAL PURDUE WORKS--ETC(U)
JUN 78 N00014-78-C-0127

UNCLASSIFIED NL

3 OF 4

AD
A068266



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

ELZER: In Germany there is an evaluation team financed by the German Ministry of Defence and organised by IABG. There is no Civil team but some organisations are interested (MBP, IDAS).

I want to organise a discussion group in Germany and have suggested that Mr. Schwald of DIN should chair it. I would try to get the group within the LTPL environment. I have promised to inform Mr. Schwald of the results of today's meeting.

There are many LTPL members already involved in evaluation groups but I have had a letter from Mr. Whitaker suggesting a separate LTPL evaluation group. (Appendix 3).

There followed a discussion about the possible dates for an LTPL evaluation group meeting and about methods for transporting material to and from the States. No conclusions were reached at this stage (see below).

THOMPSON: Unless LTPL can get a good evaluation team it may not be worth continuing with the exercise.

Material can be taken to the States via Phillip Wetherall if it's at Malvern in time.

ELZER: In my experience, you can do much more if you're at a meeting than if you merely submit the paperwork.

HOLMES: That's true if there's a substantial point we want to make but in this case it's more important to get our views down on paper as input to the evaluation review.

THOMPSON: You've already tried to get European involvement in the DoD evaluation and failed. The Commission want the LTPL-E group to keep them informed of DoD progress; it will be a side-issue if LTPL-E takes part in the review. Remember that there have been difficulties in getting funding for DoD/CEC co-operation so don't rely on funding from the CEC for a trip to the States.

LTPL must do a good job and it must give the view of European industry.

SHORTER: Is the Commission willing to support the production of a European view on the assumption that it will be kept informed of DoD activities?

THOMPSON: Yes.

SHORTER: Would the report that LTPL produces for the DoD evaluation review also do for the Commission.

THOMPSON: Yes.

SHORTER: There seems to be an additional criterion because you say you want the group to be representative of European industry.

THOMPSON: If you extract a non-representative group from LTPL for the evaluation the results could not be seen as a representative view.

HOLMES: I have a suggestion. Mr. Elzer sets up his evaluation team and sends the results to DoD. Later the full LTPL, which has representatives from national evaluation teams, can modify the report which can then be sent to CEC. I think Purdue Europe is a good time to do this because most people will be available.

SHORTER: The results could also be submitted to Purdue.

HOLMES: There's no rush. We could start the work in Zurich, the results would not be needed until the May meeting.

SHORTER: We now have a clear course of action

1. Tidy up the list of names for the LTPL evaluation team

2. Fill in the DoD application form to get the LTPL evaluation group accepted.

3. Produce a schedule of actions and a timetable for future meetings.

The Planning group then separated from the full committee to consider these points.

4. Approval of previous minutes. (Agenda item B1)

Comments should be sent to Mr. Roberts within the next three weeks.

5. Report of the Algorithmic Group

CHALMERS: Firstly, I would like to report on the activity which followed the Plenary meeting in October. On 20th and 21st October J. Barnes and I completed the draft of the current Algorithmic Proposals. This was circulated to the whole Algorithmic subgroup for comment. The only comments were editorial and the document 'Current Algorithmic Proposals', paper ref. AG 771205 was distributed to the A-list in December. Although the document does not form a complete set of proposals, it does cover all topics on which the Algorithmic Subgroup had reached agreement, listed topics which were incomplete and those which had not been discussed. It is regarded as the final document on LTPL-E Algorithmic features to be issued by the Algorithmic Subgroup, thus ending their three year period of work.

For the current meeting the Algorithmic Subgroup was asked by LTPL-C, as a result of the IPW October meeting, to collate all the Algorithmic functional requirements of LTPL-C and match them to the respective chapters of Ironman.

Yesterday the Algorithmic Subgroup met to consider this task (Note that of 7 members present only 2 were regular Algorithmic Subgroup members. Of the others 4 were established LTPL-E members from other groups (mainly Tasking) and one was a new member).

Out of a total of 16 papers listed by LTPL-C, the group had only managed to gather 8 - some of the missing ones seem to be currently unavailable within LTPL-E. Of the 8 papers, the group considered that only 5 were directly relevant, these being the "Green Sheets" and 4 LTPL-E documents. The latter had already been processed by the Algorithmic Subgroup and the results are contained in the Current Algorithmic Proposals Document.

It was decided therefore to start with a detailed comparison of the Current Algorithmic Proposals and Revised Ironman. I now invite Mr. Shorter to report on this work. *

SHORTER: We looked at the Algorithmic paper and for each point looked up the relevant Ironman sections. In this way we built up a cross-reference between LTPL and Ironman and at the same time we studied the Ichbiah, Maddock and Pyle paper in more detail than had been done before. We found some disagreements and also some areas covered in Ironman but not in the LTPL paper.

As a second exercise we went the other way, taking each Ironman point to check it against the LTPL view.

I will be writing a paper which will summarise our findings.

Our work should be useful for two reasons

1. As a significant input to the evaluation team

2. As a contribution to the ISO 'PLIP' activity via LTPL-C.

I would like the paper to go to all members of the evaluation team even if they are not on the A list.

We found some inconsistencies between the Algorithmic and Tasking proposals. At some time these two proposals need to be compared.

* After the Plenary session the Algorithmic Subgroup had a short meeting at which they quickly processed the Algorithmic part of the Green Sheets (Functional Requirements for Language Features for a Procedural Language for Industrial Computers; October 29th 1971, pages 91-93 and 96.)

CHALMERS: Should the Algorithmic group now be dissolved? Apart from one or two minor items our mainstream work is finished.

ELZER: Let's postpone this until after the DoD evaluation phase. The Algorithmic group has not produced a complete document in the way that the Tasking group has.

6. Report of the I.O. Subgroup

VERROUST: On Wednesday afternoon we had a presentation of the latest version of the current I/O technical status report (LTPL-E/335.2) and we made some minor corrections to it.

We discussed some minor points and produced an interesting proposal for a generalised formatting mechanism using the language with some new operators and types.

On Thursday we had a discussion in the presence of Mr. Kronental, a member of the Tasking Subgroup. We had two topics

1. How should we map our FILE concept onto some tasking subgroup object? Our FILE is a specific object having some properties of

   - a parallel activity

   - a task

   - a procedure

   - an object

   and the reduction of our FILE to any one of these objects leads to some difficulties.

2. What methods can be used to build a formalised model of our general FILE. We have considered the use of Petri Nets.

Our next meeting will be devoted to discussing consolidated papers on these two topics.

ELZER: I was at the meeting and felt that the File Mechanism of ALGOL 68 was worth comparing with the I.O. group's approach.

SHORTER: Have you considered whether your proposals are consistent with Ironman.

VERROUST: I have already written a paper on this topic (LTPL-E/380). The I.O. part of Ironman is very small and we agree with it although we did not agree with Tinman.

The main difference is that LTPL proposals for I.O. have features at several different levels making use of the extensibility of LTPL.

7. Report on the 'PLIP' meeting. (Agenda item B3)

ROBERT: The meeting was held on the 7th to 10th November, 1977.

The Secretary circulated copies of the "Blue Document" which he received from CCITT. This document describes a CCITT proposal for a high-level language.

We produced a paper (PLIP paper N60) on the use of acceptance criteria. Most of the work was taken from earlier papers.

Three languages were processed, all of which are candidates for international standardisation. The first was Industrial FORTRAN which has been submitted piecewise as small language extensions to FORTRAN. One document (ISA-S61.1-1976) was ready for submission to SC5. However the documents ISA-S61.2-1977 and ISA-S61.3 were found to be conflicting and more work is needed to make sure they match.

The meeting went on to consider CORAL 66 and passed a resolution concerning its submission to SC5 (Appendix 4).

There was a report on the current status of PEARL. Many firms are producing PEARL compilers although the main PEARL documents have not yet been consolidated.

The next PLIP meeting is planned for 6th November, 1978 (to be confirmed).

SHORTER: I have two additional points.

SC5 recommended that BSI build up CORAL 66 to an industrial standard.

PLIP said that they would not consider two proposals in the area of ISA-S61.2. The Germans and the Americans need to get together.

ELZER: It is not only the Germans. PLIP has contributions from many Europeans including the Purdue TC1 committee. The Americans were charged with merging all the work although there is a difference of opinion about how this is to be done.

8. Report of the Planning Subgroup. (Agenda item B2)

ELZER: Mr. Whitaker has sent me a letter saying that he would like an LTPL-E analysis group for phase I of the DoD HOL project.

The evaluation exercise has tight time-scales

March 13th	Evaluation reports needed
March 13th-27th	Review of the anlysis
March 28th - April 14th	Selection
April 15th -	Phase II

I want to be at the meeting that reviews the analysis and so I have telephoned Mr. Rummler and asked if I could be invited. He has agreed to send me an invitation.

The LTPL-E evaluation group is to meet for 1 week from 27th February to 3rd March. Those who can are invited to come and travelling expenses will be paid by CEC.

The best way to get the documents from DARPA is via an RAF flight. There are two possibilities for getting documents back

1. via an RAF flight

2. to take them myself.

Mr. Thompson would like a consolidated view of all European evaluation groups. We are therefore planning to do two things.

1. Present our evaluation to Purdue and initiate a discussion.

2. Write to all members of the European evaluation teams and invite them to our next meeting in May. There are to be two letters, one from us containing an invitation and one from CEC encouraging people to accept.

I need a complete list of people who will be attending the LTPL evaluation group by 31st January.

MALAGARDIS: A united European view will be much more important than views from national bodies.

SHORTER: A European view is important but smaller evaluation groups are stronger. However, it is important to give a European view to the CEC.

ELZER: At Purdue there will be a session devoted to DoD work. This will include the following items:

1. A presentation of LTPL's results

2. A discussion

3. A vote.

MALAGARDIS: We will find someone from the contractors or the DoD to give a presentation.

9. <u>Preparation for the Purdue Spring Meeting. (Agenda item B5)</u>

ELZER: Invitations to LTPL-E members have been delayed but will be distributed in the near future.

LTPL's contribution to Purdue will be the presentation of the DoD evaluation.

10. Meeting Structure etc. (Agenda item B6)

ELZER: The mailing list has been purged.

I will distribute an up-to-date version of the paper list.

11. Next Meetings. (Agenda item B7)

Zurich : April 4th to April 7th (Purdue Spring Meeting)
Brussels: May 31st to June 2nd.

12. Any Other Business. (Agenda item B8)

ELZER: We must decide what happens after the DoD work. There are two possible areas of work.

1. Program production (structured programming etc.)

2. Monitoring DoD.

HOLMES: An additional one is to specify an environment for DoD that would make it suitable for industrial applications.

MALAGARDIS: I suggest that Mr. Holmes writes a paper and presents it at Zurich.

ELZER: I think that is too early.

MALAGARDIS: There is time at Zurich for a discussion.

ELZER: That is O.K. if you don't force a decision.

CHALMERS: It is important to show other Technical Committees that LTPL has some new objectives after the DoD work.

HOLMES: It is also important to show the CEC.

MALAGARDIS: I have told the Purdue Europe executive committee that we need an overall advertisement of Purdue Europe activities. I have been approached by a Swiss editor to do this. The idea is to publish working papers of Purdue in a way similar to the SIGPLAN notices.

The publishers are called Georgi and they already produce a journal called "Digital Process Review".

ELZER: From all the enquiries I have received, I believe that we could have sold our language comparison document.

MALAGARDIS: Until the CEC have a mechanism for publishing our papers they cannot help us. I have told the publishing house that the arrangement cannot be permanent.

SHORTER: It would be possible to give the CEC some return for financing our meetings.

FROGGATT: The Tasking Subgroup has already sent its proposals to "Software Practice and Experience".

Meeting Closed.

APPENDIX 1 TO MINUTES OF LTPL-E 40TH MEETING

FROM: MINDEF

SUBJECT: MORE ANALYSIS

TO: HOLWG

CC: WHITAKER, WETHERALL, MINDEF

FURTHER TO OUR MSG OF 23 DEC 77 OUTLINING PROPOSED ADDITIONAL UK ANALYSIS TEAM, IT IS UNDERSTOOD THAT LTPL(E) IS ALSO PLANNING TO RESPOND. IT IS BEING SUGGESTED THAT THEY HOLD A ONE WEEK MEETING IN FEB/ MAR 78. MINDEF WILL PROVIDE LIMITED SUPPORT SUCH AS ARPA NET FACILITIES, CARRIAGE OF DOCUMENTS IF REQUIRED. FURTHER DETAILS WILL FOLLOW WHEN THEY ARE AVAILABLE.

NICK NEVE

APPENDIX 2 TO MINUTES OF LTPL-E 40TH MEETING

339747 MOD PEV G
22777 GECBWD G
FILE NO 670   18.1.78 DMC

TO MR N NEVE

PLEASE SEND FOLLOWING MESSAGE ON ARPANET TO HOLWG FOR WHITAKER AND WETHERALL.

"FURTHER TO OUR MESSAGE OF (MR NEVE TO INSERT DATE) HEADED "MORE ANALYSIS" LTPL-E HAVE NOT YET SUCCEEDED IN ARRANGING ANALYSIS TEAM. THE SUBJECT WILL BE DISCUSSED AT LTPL-E MEETING 25TH TO 27TH JANUARY WHERE DECISION WILL BE MADE, TEAM NOMINATED ETC., THE NECESSARY DETAILS WILL BE SENT TO YOU IN WEEK COMMENCING 30TH JANUARY.

MEANWHILE CORRESPONDENCE SHOULD BE SENT TO MR A F CHALMERS, GEC COMPUTERS LTD, ELSTREE WAY, BOREHAMWOOD, HERTS, ENGLAND. (TELEX 22777)".

REGARDS ALAN CHALMERS SPECIAL PROJECTS EXECUTIVE.

339747 MOD PE G
SENT 1028 DMC

APPENDIX 3 TO MINUTES OF LTPL-E 40TH MEETING

ADVANCED RESEARCH PROJECTS AGENCY

U.S. Jan. 1978

MEMO FOR Peter Elzer

I am making up a list of analyzers for the phase I products. I have a number of LTPL-E members on this already but reckon that we ought to make a more official show for LTPL-E as an organization rather than just individuals. Rummler suggests that we make you, Timmesfeld and Verroust the contacts and send the reports there first. The time span is short and I realize that resources are limited, but if you are "out of school" by then we would value your contribution. Whether you all want to come over or not is up to you, probably something you should reserve until you see how things look. Actually I sort of hope that the answers will be fairly straight forward and can be best expressed in writing, but maybe I am too optimistic. In any case we want a permanent record of the analyses so that the continuing contractors can benefit from them. This is the next-to-last chance to infulence the design, and probably the last chance for significant impact.

I am sending, for your random information, the current HOLWG ARPANET message file. There are some items which may be of interest.

Bill Whitaker

APPENDIX 4 TO MINUTES OF LTPL-E 40TH MEETING

Resolution 9

Whereas CORAL 66 was referred to WG 1 as an initial project by ISO/TC 97/SC 5 in 1975,

whereas the WG 1 has reached consensus that CORAL 66 lies within its scope, WG 1 has also concluded that the unique attribution of CORAL 66 make its scope of applicability much greater than only Industrial Real-Time systems.

Be it resolved that ISO/TC 97/SC 5/WG 1 suggests that the consideration of CORAL 66 by SC 5 would be appropriate and timely. It further suggests that an appropriate body be charged with the processing of the CORAL 66 proposal (N 20).

It is further resolved that, on the assumption that SC 5 arranges for the processing of CORAL 66, WG 1 would welcome the timely submission of a specific proposal containing explicit Industrial Real-Time features intended for use with CORAL 66.

15 FOR - 0 AGAINST - 1 ABSTAINS

LTPL - EUROPEAN GROUP

Author: A.F. Chalmers
M. Kronental
D.N. Shorter

Institution:

LTPL - E / ACMKDS 780404

Category: B

Updates: None

Replaces: LTPL-E/AC780403

pp: 5

Date (assigned): 4th April 1978

Date (completed): 4th April 1978

Title: Proposals for the way ahead for LTPL-E

Contents:

Notes:

(1) These proposals include the ones made by Mr. G.W. Holmes in paper LTPL-E/GH780206

(2) The proposals in this paper were agreed by the LTPL-E Committee Meeting at Zurich on 4th April 1978 and form the justification and basis for a revised work programme

PRECEDING PAGE BLANK - NOT FILMED

PROPOSALS FOR THE WAY AHEAD FOR LTPL-E

1. INTRODUCTION

Given that the CEC has not been able to obtain the agreement of the Member States for the LTPL-E Project Plan, coupled with the recognition of the DOD-HOL as an on-going programme we have two good reasons for deciding a changed role for the LTPL-E committee. There are aspects of the DOD-HOL programme itself which justify attention, together with the whole question of a European language and program development system for industrial control and automation. Whether it is based on DOD-HOL or not has in itself to be investigated. The body of expertise within LTPL-E has also the capability of looking further ahead.

This paper is a first attempt to set out objectives and a broad work programme to be started on now. This may well change after say one year when DOD Phase 2 is evaluated from a European angle.

2. RAISON D'ÊTRE OF TC3

2.1. The objectives and the work programme must have the prime purpose of continuing to serve the language needs of the European community of Industrial Users, Manufacturers etc., involved in Real Time Control and Automation.

2.2. Advising DG III of the CEC and the Government Departments of the Member States. In particular links with the latter should be more positively established to and from LTPL-E.

2.3. It is necessary to:-

(a) Review current practices and techniques in use in high level languages in the industrial environment with a view to advising or assisting in making improvements.

(b) Look ahead to future requirements and in particular to take account of evolving technology, in both hardware and software areas, so as to develop and adapt our longer term ideas.

3. PRIMARY TECHNICAL OBJECTIVE

The prime objective should be the establishment of a real-time industrial control language for Europe, with the necessary systems and user aids.

This objective should take proper account of the following:-

(a) The on-going DOD-HOL programme for embedded computer systems (the most significant influence on our work)

(b) The outcome of LTPL-E work to date (most of the good technical work has been completed or is in a clearly defined state).

(c) The lack of acceptance by the European community of the LTPL-E proposals for an LTPL project plan.

(d) Progress on development, user acceptance and standisation of:

  (i) national language activities within Europe, e.g. Coral 66, RTL/2, LTR and PEARL

  (ii) those of related bodies (European or world-wide) e.g. CCITT (CHLL?), Process Fortran, Real Time Basic and Real Time Operating Systems.

The work of RTTG of ECMA on this subject should be investigated.

4. LIAISON ACTIVITIES

4.1. Continue, but formalise and strengthen the LTPL-E to DOD-HOL liaison

4.2. Arrange, if possible, to co-ordinate, or centralise through LTPL-E, the various national (industrial) links with DOD, and with CEC. A positive suggestion is a strong two-way communication between the Senior Official(s) of the Member States and LTPL-E (through a nominated LTPL-E member) for each appropriate state. Note: This link should apply to all LTPL-E work.

4.3. Encourage CEC, with Member State support, to formalise a link with DOD with the purpose of guaranteeing timely access to DOD-1 material, with the freedom to adapt it and control it locally for European Industrial needs.

4.4. Maintain TC3's existing IFAC and IFIP links through TC3's continuing membership of Purdue Europe and hence within the LTPL-C group of the International Purdue Workshop.

4.5. Maintain the necessary liaison with DG III of the CEC and with the various Government departments of the Member States.

4.6. Monitor the necessary liaison with ISO, PLIP and with the various national standards bodies.

5. SPECIFIC ACTIVITIES IN PRIORITY ORDER

5.1. Test cases and examples.

Establish appropriate test problems for the evaluation of languages and maybe compilers.

5.2. Present the US DOD HOL work to European Industry.

Set up a forum to present this work to allow assessment by potential users. Initially this forum will emphasise the European evaluations of the two phase 1 language proposals selected by DOD for phase 2 development.

5.3. Industrial requirements for system development and running aids. Monitor the DOD environment work to ensure it includes:-

(i) Program development tools - editors, windowing techniques for debugging, object computer compilation.

(ii) Support modules including industrial library modules

(iii) Operating system primitives

This work should aid the decision on whether to adopt DOD-1 after Phase 2, to subset it, superset it, alter it or whatever. This is a fundamental and important question and it will be easier to arrive at a rational technical decision if the ground-work is done.

5.4. 'LTPL-E Extraction'

Identify, examine and correlate the latest significant technical work of LTPL-E e.g. The Tasking Proposals, principles (not detail) of the Algorithmic Proposals. Continuation of the I/O work should now take account of the DOD 1 I/O features and those of the European industrial requirement. The work programme will be considered by the I/O Subgroup at the next LTPL-E meeting, taking account of the current LTPL-E I/O work reported in paper 350/2.

5.5. Review of Current Practices and Techniques

Study current practices and techniques with a view to recommending improvements and providing advice to potential users of existing languages, taking account of the need to lead into a new language. This should probably be background activity at present, in that 5.1 - 5.4 are of prime importance at the moment.

5.6. 'Look Ahead' Activities

The LTPL-E committee must remain alive to the longer term future, possibly stepping up this activity in later years. This will involve study of future requirements, examination of evolving hardware technology and its likely impact on future industrial systems, and looking at evolving (ultra) high level language and programming techniques. It has been suggested that even longer term work should cover non-procedural languages.

APPENDIX E-IX

TC-5

INTERFACES AND DATA TRANSMISSION COMMITTEE

PURDUE EUROPE

1. Agreed Upon Specifications of SIRE, April 1978.

PRECEDING PAGE BLANK - NOT FILMED

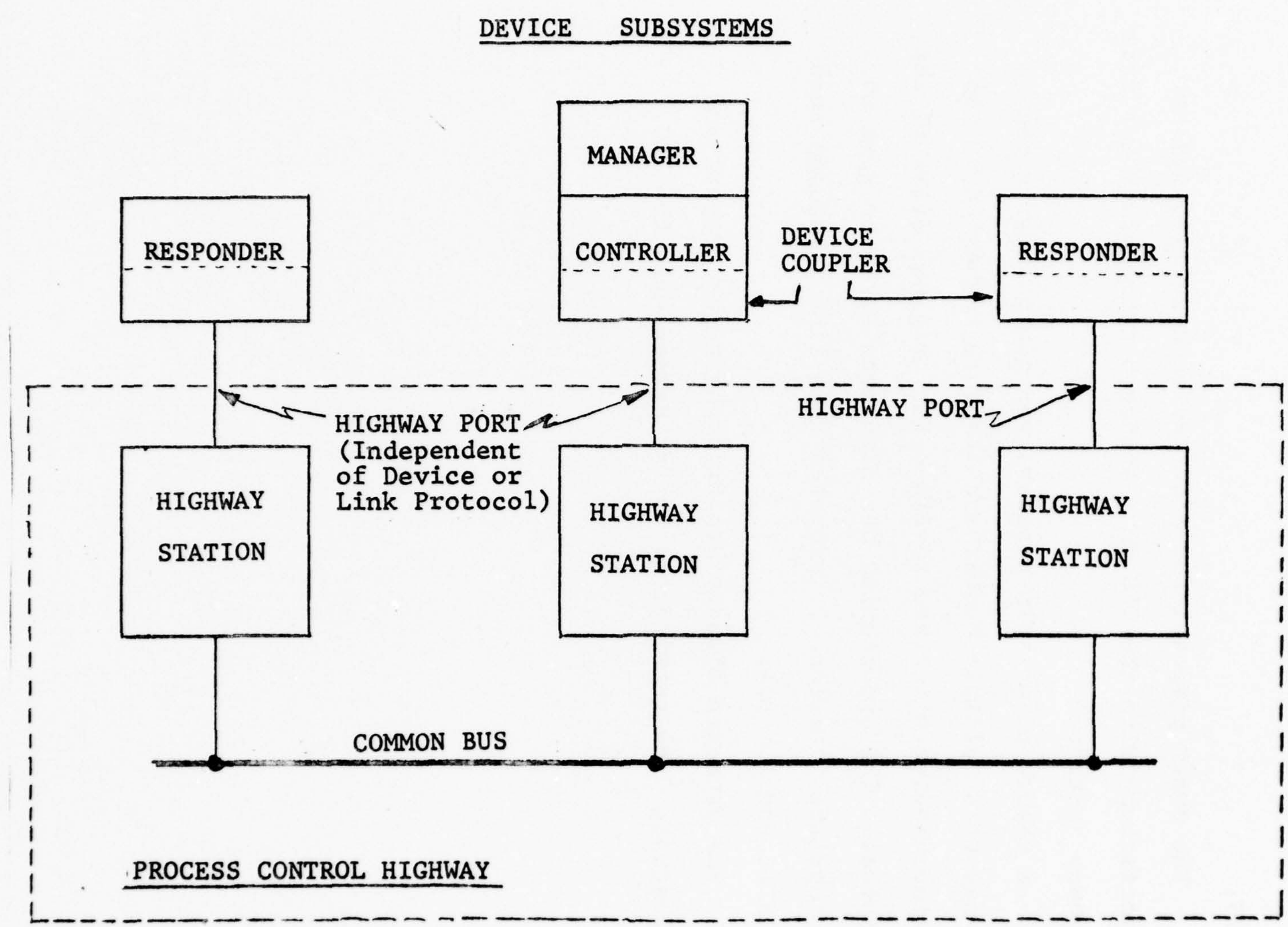
AGREED UPON SPECIFICATION OF "SIRE"

TC-5.

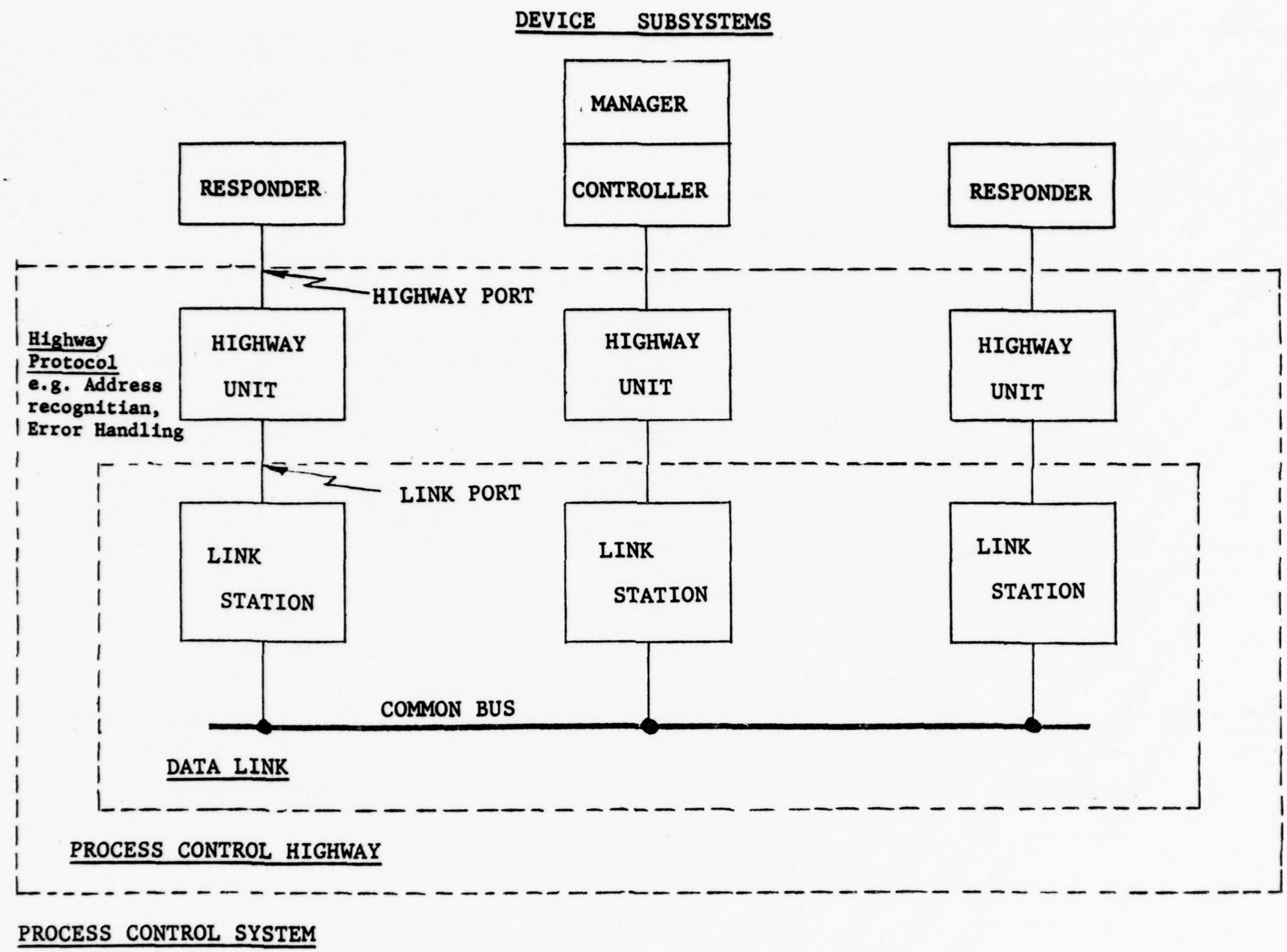
The draft proposal for SIRE has been delayed due to negotiations with the IEC on a standard terminology for data highway systems.

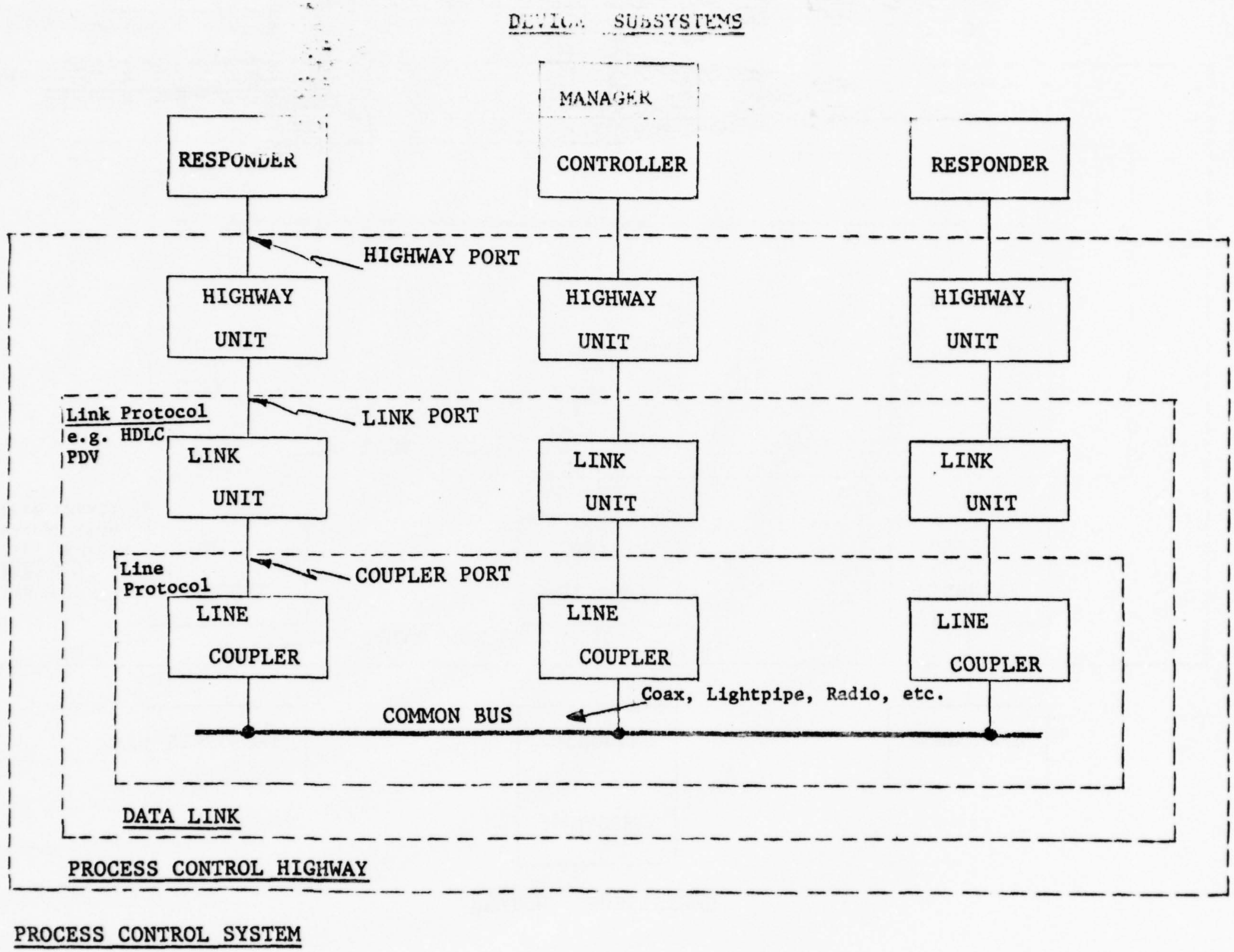
A report on work in progress was given showing the change to a multiple Master architecture requested by the various bodies which had reviewed the original draft of the proposal. The opportunity has also been taken to give an appropriate interface to proposed date link standards such as HDLC and PDV bus.

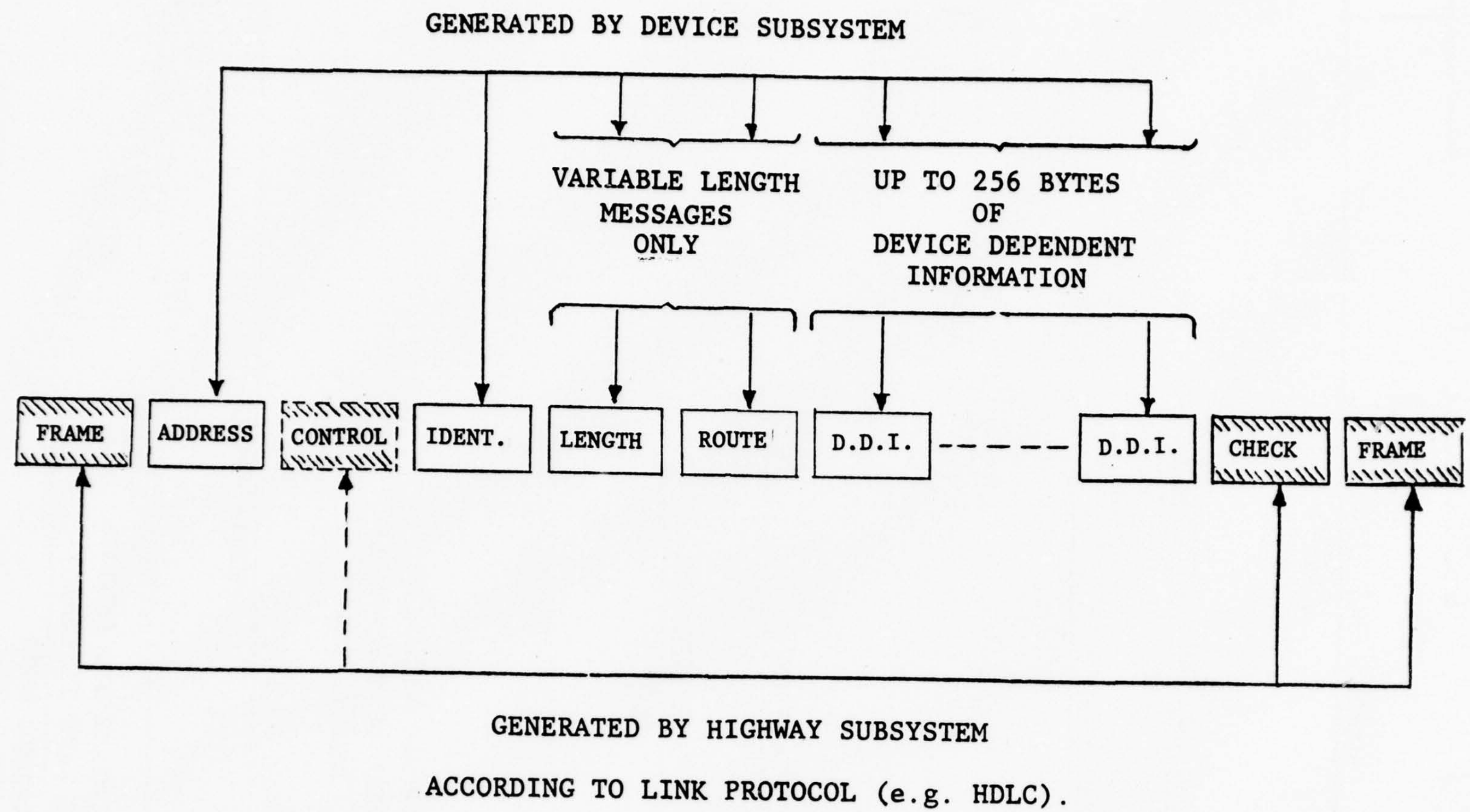
The attached figures are from the verbal report and illustrate the terminology so far agreed with the IEC.



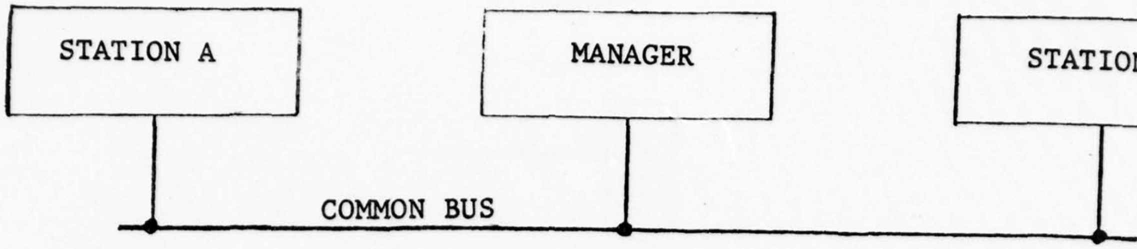
PROCESS CONTROL SYSTEM

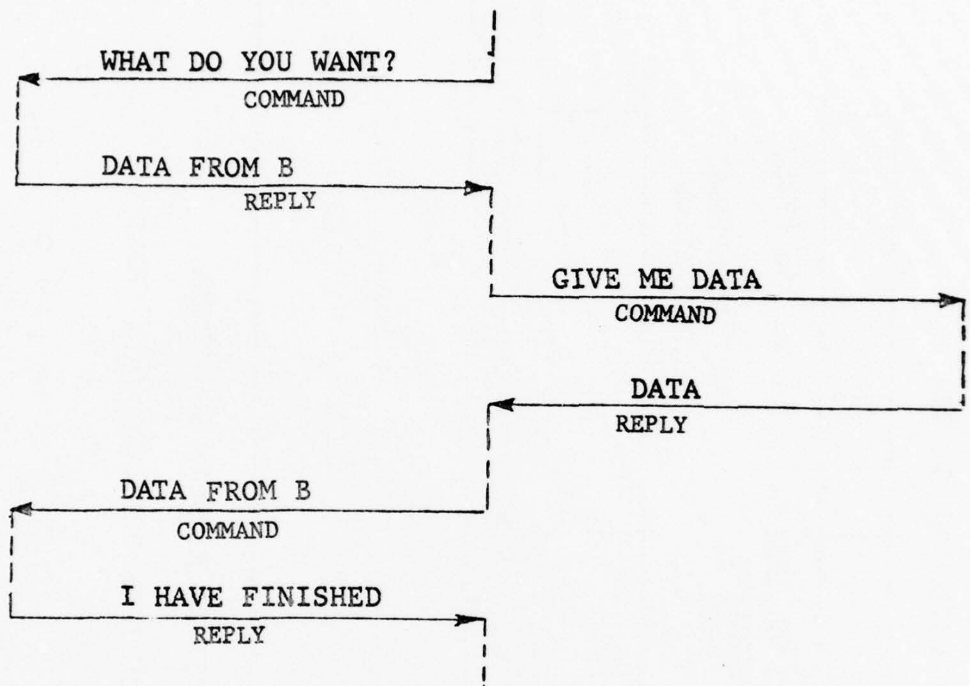

DEVICE SUBSYSTEMS
MANAGER
CONTROLLER
RESPONDER
RESPONDER
HIGHWAY PORT
Highway Protocol e.g. Address recognitian, Error Handling
HIGHWAY UNIT
HIGHWAY UNIT
HIGHWAY UNIT
LINK PORT
LINK STATION
LINK STATION
LINK STATION
COMMON BUS
DATA LINK
PROCESS CONTROL HIGHWAY
PROCESS CONTROL SYSTEM


DEVICE SUBSYSTEMS
MANAGER
RESPONDER
CONTROLLER
RESPONDER
HIGHWAY PORT
HIGHWAY UNIT
HIGHWAY UNIT
HIGHWAY UNIT
Link Protocol
e.g. HDLC
PDV
LINK PORT
LINK UNIT
LINK UNIT
LINK UNIT
Line Protocol
COUPLER PORT
LINE COUPLER
LINE COUPLER
LINE COUPLER
Coax, Lightpipe, Radio, etc.
COMMON BUS
DATA LINK
PROCESS CONTROL HIGHWAY
PROCESS CONTROL SYSTEM

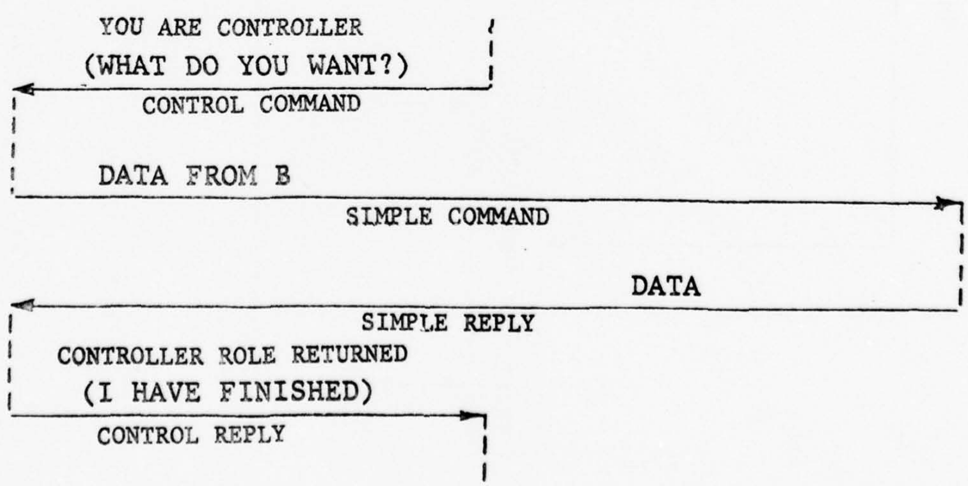


MESSAGE STRUCTURE

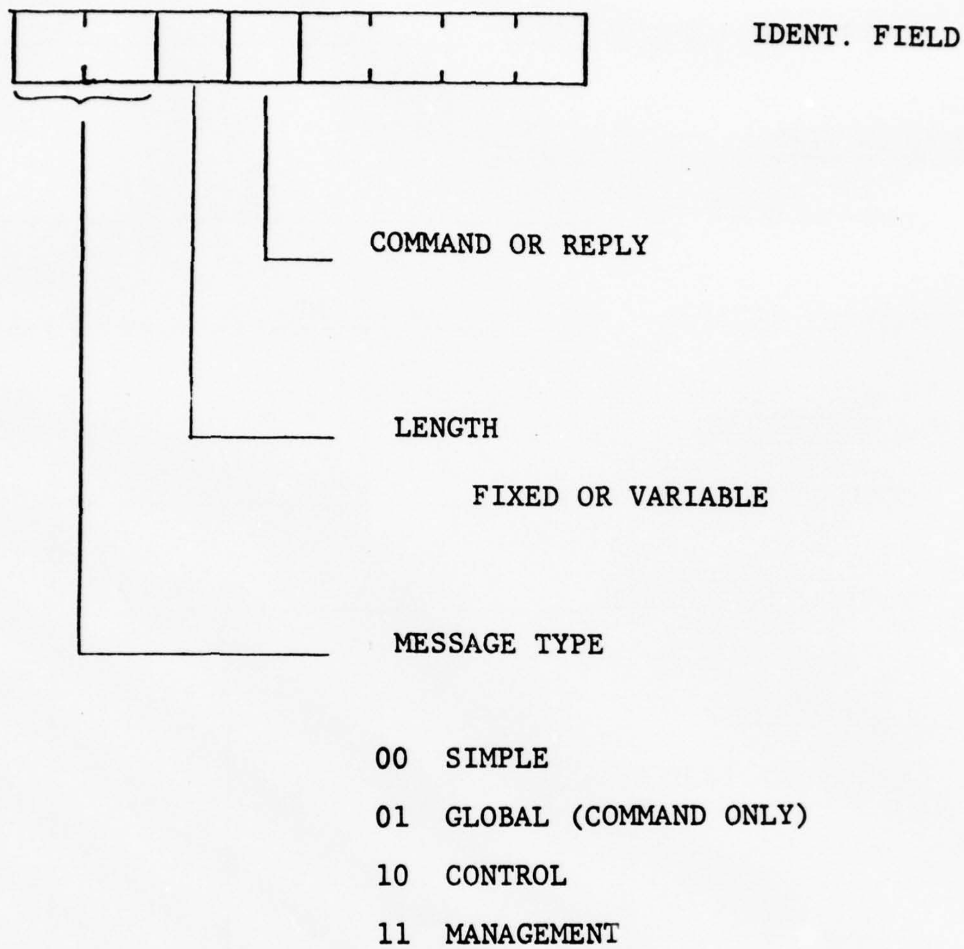




1. MANAGER RETAINS CONTROLLER ROLE

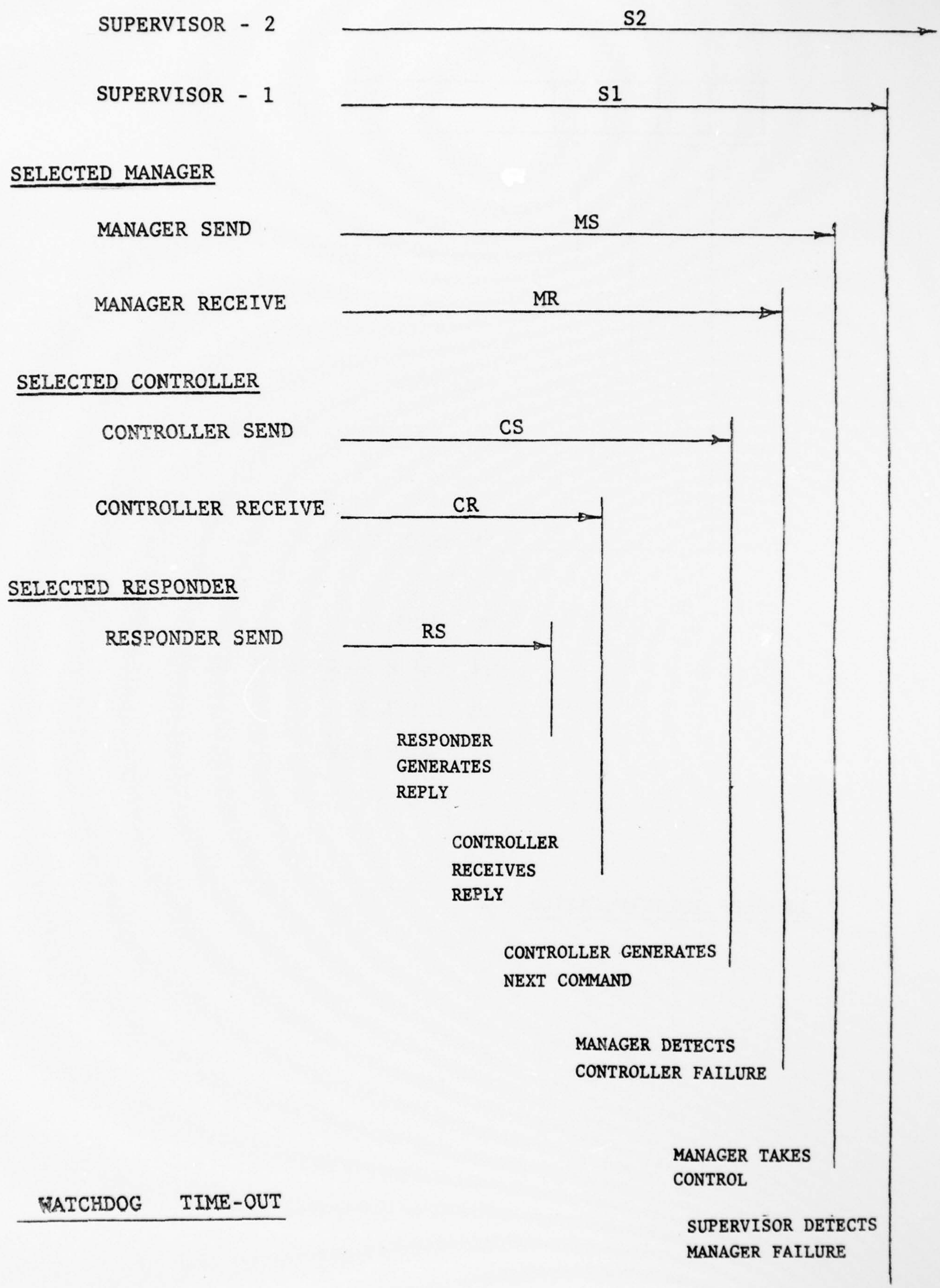


2. MANAGER DELEGATES CONTROLLER ROLE



MESSAGE IDENTIFICATION

STANDBY MANAGER



APPENDICES E-X

TC-6

MAN/MACHINE COMMUNICATIONS COMMITTEE
PURDUE EUROPE

1. Aune, A. B., An International Activity for Enhancement of Man-Machine Communications Design in Industrial Computer Control Systems, Holden Programme Group Meeting, Loen, Norway, June 5-6, 1978.

2. Reasons for Implementing and Not Implementing Modern Man-Machine Interface Functions.

AN INTERNATIONAL ACTIVITY FOR ENHANCEMENT OF MAN-MACHINE COMMUNICATIONS DESIGN IN INDUSTRIAL COMPUTER CONTROL SYSTEMS

by

Arthur B. Aune
SINTEF, Division Automatic Control
N-7034 Trondheim, Norway

ABSTRACT

The paper presents the work of the Technical Committee on Man-Machine Communications of the International Purdue Workshop on Industrial Computer Syste
This organization have regional branches in America, Japan and Europe. The European branch, PURDUE-EUROPE, is partly sponsored by the Commission of the European Communities.

The committee is concerned with the functioning of the human being as a part of complex industrial systems. This functioning should satisfy both human wellbeing and cost-effectiveness.

The present work of the committee is divided in three areas:

- collect and evaluate human factor (ergonomic) information, experience, ideas and methodology relevant to man-process interaction and man-machine systems design.
- to disseminate results of these studies.
- to produce guidelines for design, implementation, training, etc.

The paper also presents interactions with other international or national activities relevant to the work of the committee.

OECD HALDEN REACTOR PROJECT. MEMBERS { A, USA, JAPAN, FRG, I, NL, DK, N, S, SF

Paper to be Presented at the Enlarged Halden Programme Group Meeting
Loen, Norway, 5th - 9th June, 1978 Attendance ≈ 260. engineers

SINTEF The Foundation of Scientific and Industrial Research at The University of Trondheim, Norway.

1. INTRODUCTION

1.1 The International Purdue Workshop on Industrial Computer Systems

The International Purdue Workshop on Industrial Computer Systems, in its present format, came about as the results of a merger in 1973 of the Instrument Society of America (ISA) Computer Control Workshop with the former Purdue Workshop on the Standardization of Industrial Computer Languages, also cosponsored by the ISA. This merger brought together the former workshops' separate emphases on hardware and software into a stronger emphasis on engineering methods for computer projects. Applications interest remains in the use of digital computers to aid in the operation of industrial processes of all types.

The new combined international workshop provides a forum for the exchange of experiences and for the development of guidelines and proposed standards throughout the world.

Regional meetings are held each spring in Europe, North America and Japan, with a combined international meeting each fall at Purdue University. The regional groups are divided into several technical committees to assemble implementation guidelines and standards proposals on specialized hardware and software topics of common interest. Attendees represent many industries, both users and vendors of industrial computer systems and components, universities and research institutions, with a wide range of experience in the industrial application of digital systems.

The International Workshop is sponsored by the International Federation of Information Processing (IFIP), the International Federation of Automatic Control (IFAC), the Instrument Society of America (ISA), the Purdue Laboratory for Applied Industrial Control of Purdue University, and other national and international bodies.

Its objectives are:

To make the definition, justification, hardware and software design, procurement, programming, installation, commissioning, operation, and maintenance of industrial computer systems more efficient and economical through education, the organisation and interchange of information, and the development of standards and/or guidelines.

An organizational scheme of the International Purdue Workshop on Industrial Computer Systems is shown in Figure 1.

1.2 Purdue-Europe

PURDUE EUROPE is the regional organization of the International Purdue Workshop. It is structured in Technical Committees, each being chaired at the local level by a responsible person competent on the topic.

PURDUE EUROPE is sponsored by the Commission of the European Communities, Directorate General for Internal Market and Industrial Affairs (DG III).

At present the following committees are active:

Industrial Real Time FORTRAN (TC1),
Industrial Real Time BASIC (TC2),
Long Term Procedural Languages (TC3),
Problem Oriented Languages (TC4),
Interfaces and Data Transmission (TC5),
Man-Machine Communications (TC6),
System Reliability, Safety and Security (TC7),
Operating Systems (TC8),
European Distributed Intelligence Study Group (TC9)

1.3 The Man-Machine Communications Committee

This committee is concerned with the functioning of the human being as a part of complex industrial systems. This functioning should satisfy both human wellbeing and cost-effectiveness.

The objectives of the committee are

- to collect and evaluate
  - o user experience and ideas concerning the man-process interaction
  - o human factors information (sources and results) relevant to man-machine systems
  - o information on the "state of the art" in available implementation tools (display/controls technology, dialogue-based software, etc.)
  - o methodology and techniques for man-machine system design
- to disseminate relevant results of these studies to encourage discussion with and acceptance by users.
- to produce guidelines in various areas such as, fx., design, implementation, training, etc.

The results will be disseminated as follows:

- in working papers (to be published by the authors in communication of the Purdue organization or in technical journals) which have been discussed by the TC but which may not represent the final opinion of the TC.
- in guidelines which are the official representations of the TC.

Guidelines will be presented to national and international professional groups and eventually to standardization bodies.

At present, the committee has about ten active members, representing various sectors of european industry, universities and national laboratories. whose special interests include

- the process operator
- the software aspects of the man-machine interface
- evaluation of display systems and control consoles
- continuous production systems
- interactive system design
- display coding
- safety and security

and the following countries are represented

- Denmark
- Federal Republic of Germany
- France
- England
- Norway
- Sweden
- Switzerland
- The Netherlands

2. SCOPE OF WORK

2.1 Trends in the design of industrial computer systems

There has been a gradual development in high-level automation of industrial production processes, starting in electrical power generation, petroleum and chemical sectors and spreading to, amongst others, the steel industry, assembly operations and public transport.

During the last few years, however, developments within the scope of industrial computing exercise a considerable influence. A variety of, devices, such as microprocessors and visual display units, are now available for industrial and non-industrial use. Their costs, compared to the cost of labour, have become so low that wide spread application for enhancing productivity is unavoidable.

This dramatic development of technology has created a situation in which automation systems are radically different for new plants. New designs are being introduced without a thorough knowledge based on prior experience. This is clearly apparent in the application of multiple cathode-ray tube displays for process supervision and the use of distributed computing networks. A fundamental gap is emerging between the revolutionary rate of change in machine technology, and the incorporation of human factors. These encompass the existing knowledge about human skills, proven methods for analysing human performance and the understanding of human attitudes towards automation.

The problem is aggravated by the upgrading of plants and the increasing complexity of processes owing to energy and material recycling. Narrowing profitability margins put a heavy emphasis on production within tolerance limits and on coping effectively with abnormal conditions. The safety

aspect of plant operation is also gaining importance, due to legislation in an age of growing public awareness of and concern with the environment in the widest sense.

The operator, whose main task is now to supervise process operation, has a key role in this development. His operational practices and procedures are strongly influenced by the design of the man-machine interface (MMIF). Careful design, considering environmental, operational, management, legal and social requirements, is therefore crucial to the success of the man-machine system (See Figure 2). But conflict can arise between the natural tendency to assign as much responsibility and control to machines as technologically possible against the need to enable the human operator to maintain meaning in an understanding of his job - to provide for his interpretive skills in decision making and for his ultimate responsibility for plant control and for plant safety.

In some respects, human performance is poorer than that of a machine, e.g. in making consistent decisions of calculations. But design for complementary functions of human operators and equipment can exploit the best features of both. In no industry is this more apparent than in that of nuclear energy where risk analysis, safety reporting and incident analysis are becoming increasingly important. Many studies have been undertaken investigating human fallibility and means of allowing for it through good design practice. These considerations are equally appropriate to the many other process industries for systems design to be safe, acceptable in operational requirements for humans, and to be cost effective.

2.2 Work objectives

These various factors mentioned above demonstrate the need for:

- design procedures which contribute to acceptable tasks for human operators
- techniques in designing interfaces to match human abilities
- utilisation of information describing human performance

The following means will be used:

- Survey of successes and failures in the design of MMIF,
- Determination of the parameters and features critical for good MMC design and the incorporation of these in a series of guideline documents, for example on the methodology and techniques for design, implementation, and for training,
- Cooperation with other TCs in matters concerning MMC at all levels of system design, implementation and use,
- Dissemination of the results, encouraging discussions with and acceptance by users, national and international groups and standardisation bodies.

2.3 Past and current activities

Till now the following activities have been carried out

- Preparation, in cooperation with the parallel American TC, of a set of guidelines which offer procedural and substantive recommendations. The guideline is based on the systems design flowchart shown in Figure 3.

  Publication: GUIDELINES FOR THE DESIGN OF MAN-MACHINE INTERFACE FOR PROCESS CONTROL,
  International Purdue Workshop, 1976.

- Participation of the results from a questionnaire which has been circulated to various European industries to assess experience and attitudes to problems in interface design, from basic philosophy to realisation. An analysis of the common problem areas is used to formulate a working remit based on users' needs.

  Publication: THE OPERATOR INTERFACE IN PROCESS CONTROL IN EUROPEAN INDUSTRIES,
  Purdue-Europe, 1977.

Furthermore, several new activities have started

- Establishment of contacts with industry, national laboaroaries and universities engaged in the field of MMC throughout Europe. A survey on "Man-Machine Research in Europe" has been initiated.

- In order to further a growing awareness of and to promote active interest in "modern man-machine systems design", the committee plans to organize seminars, workshops, etc. for industrial users and for representatives of national bodies.

  The first attempt in this matter has materialized in two Round Table Discussions at the IFAC Congress 78 in Helsinki, June 1978.

  RT11 INTERDISIPLINARY APPROACH TO AUTOMATED SYSTEMS DESIGN
  RT13 INDUSTRIAL COMPUTER SYSTEMS AND MAN-MACHINE COMMUNICATIONS

  A committee workshop for discussion of guideline developments with representatives of european industries is scheduled for December 1978 in Brussels.

- To formalise co-operation with The Commission of European Communities (CEC) to increase interest and acceptanve of the committees work in European industry.

- A joint activity with the parallel American committee is in progress, in order to develop a new set of general guideline documents. It is planned that their structure will be suitable for providing information to senior management, project personnel and users as appropriate.

3. A GENERAL GUIDELINE CONCEPT

The European Committee on MMC feels the need to supplement the previous mentioned Guideline of 1976 with material aimed at the initial design stages. The questions that may be asked concerning man-machine systems cover a wide range of disciplines and it would be impossible to provide all relevant information within the compass of a single document. The principle adopted is to provide check lists for the design teams. It is necessary to consider all check list items for relevance, and to follow up those which are relevant. On items where brief unambigious advice cannot be supplied, additional information sources (bibliographies, review documents based on experience

from R&D in interesting areas, etc.). The basic framework of such a set of guidelines is indicated in Figure 4.

The guidelines will be structured at 3 levels, which corresponds to:

- 1st level - WHAT - policy and strategy of the overall design process
- 2nd level - HOW - design procedure
- 3rd leved - DOING IT - design

and the contents are directed primarily to the groups responsible for carrying out the work at those levels.

Level 1 is directed to those responsible for the primary decisions on how the system is to be acquired, and for setting its basic outline and parameters. It deals with those aspects of the major decisions which can influence the nature and form of the man-machine interface, including the level of automation which is aimed for, use of ergonomic expertise, etc.

Level 2, which depends on the level 1 decisions, covers the analysis of the major functions of the plant and control system, their allocation to man and machine and leads to the specification of the overall operational procedures and interface facilities. This design level is influenced by many human and technological factors.

Level 3, which depends on the level 2 decisions, is concerned with the detailing of the interface in terms of specified operational tasks and console equipment etc. Finally advice is given on methods for evaluating the man-machine interface.

Many of the issues that affect the design of man-machine systems are "process independent". Consequently, much of the material in the guidelines will be applicable over a wide range of control system applications. This is not to say that a "process independent" question necessarily has a "process independent" answer, and in cases where it does not, all that can be done is to indicate the source of additional information specific to the type of application (Fig. 4).

The term "process" is to be interpreted broadly. In many cases a process is tied to localised plant, as for example a petrochemical or cement plant. It can however be interpreted to cover other control situations, for example traffic control of various sorts, and even situations which do not have a

real-time aspect, such as information systems. Although the guidelines are biased towards industrial plant control, it is hoped that they will contain matter which is useful to interface designers in other situations as well.

4. INTERACTION WITH OTHER ORGANIZATIONS

In the pursuit of its objectives the Committee seeks to utilize information, results and experiences coming from many sources

- committee members own work
- other committees of the International Purdue Workshop
- various international and national organizations
- other technical organizations (e.g. IFAC, IFIP)

Examples on the Committees efforts to seek for and utilize information from various sources, are (Fig. 5)

- Through committee members to observe relevant activity of IEEE/IAEA, nuclear reactor research project (OECD-Halden) and CERN.
- Through committee members to observe relevant activity of the FRG research project (PDV) on Computerized Process Control.
- To co-operate with the International Standardization Organization committee, ISO/TC159 on Ergonomics and Standards.
  This committee has five sub-committees, with two of them having relevance to MMC committees present work

  SC1: Ergonomic guiding principles
  SC4: Signals and controls

The Man-Machine Communications Committee of the International Purdue Workshop hopes that it can benefit from all information sources, shown in Figure 5, in order to provide usefull tools for design and implementation of future Man-Machine Communication Systems of various operational situations.

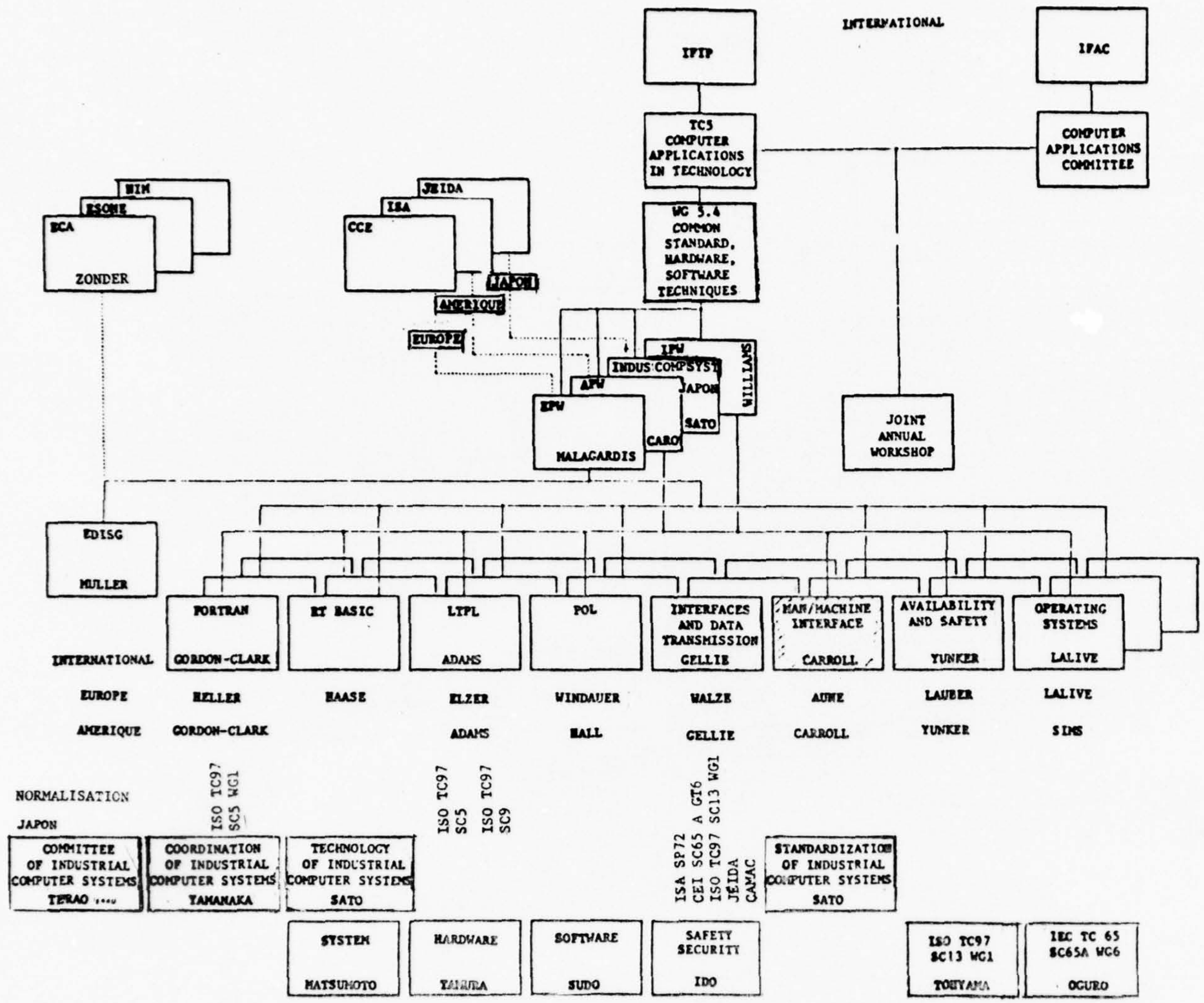


Fig. 1. Organizational Scheme of the International Purdue Workshop on Industrial Computer Systems (IPW). Regional branches and sponsoring organizations.

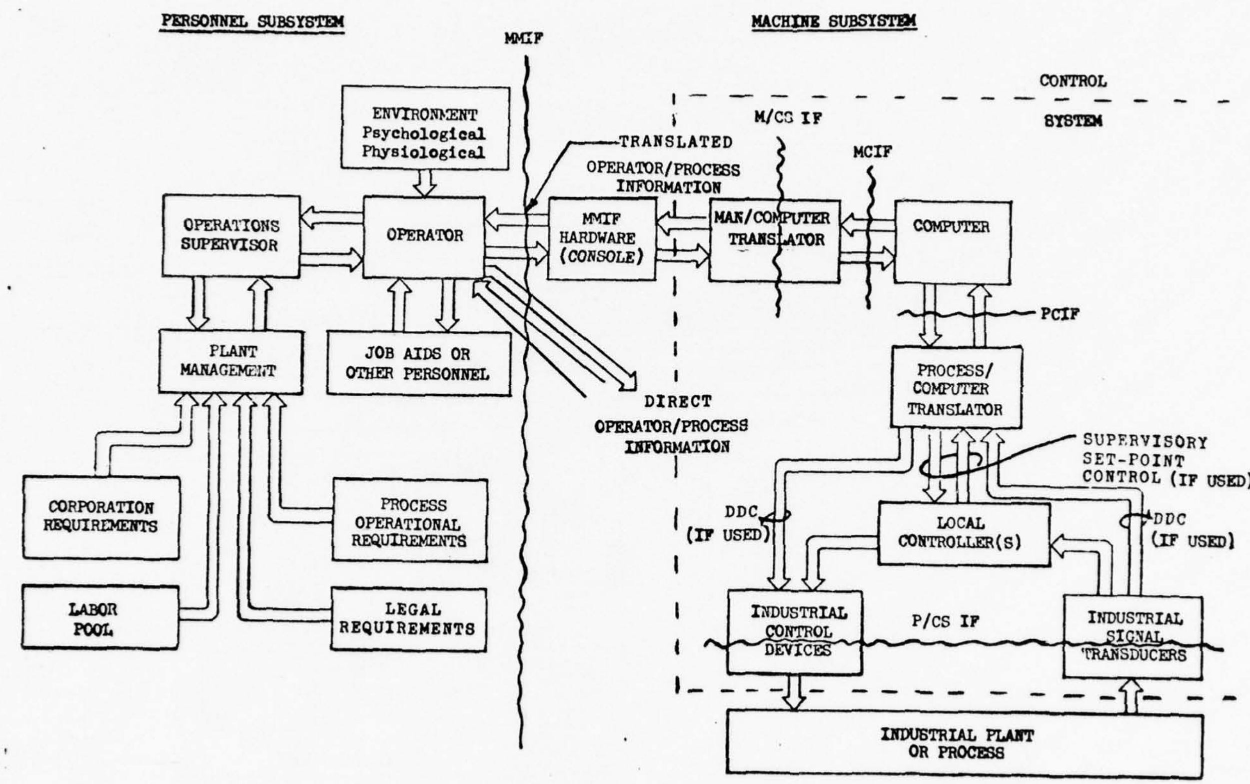


Fig. 2. The Man-Machine System Model

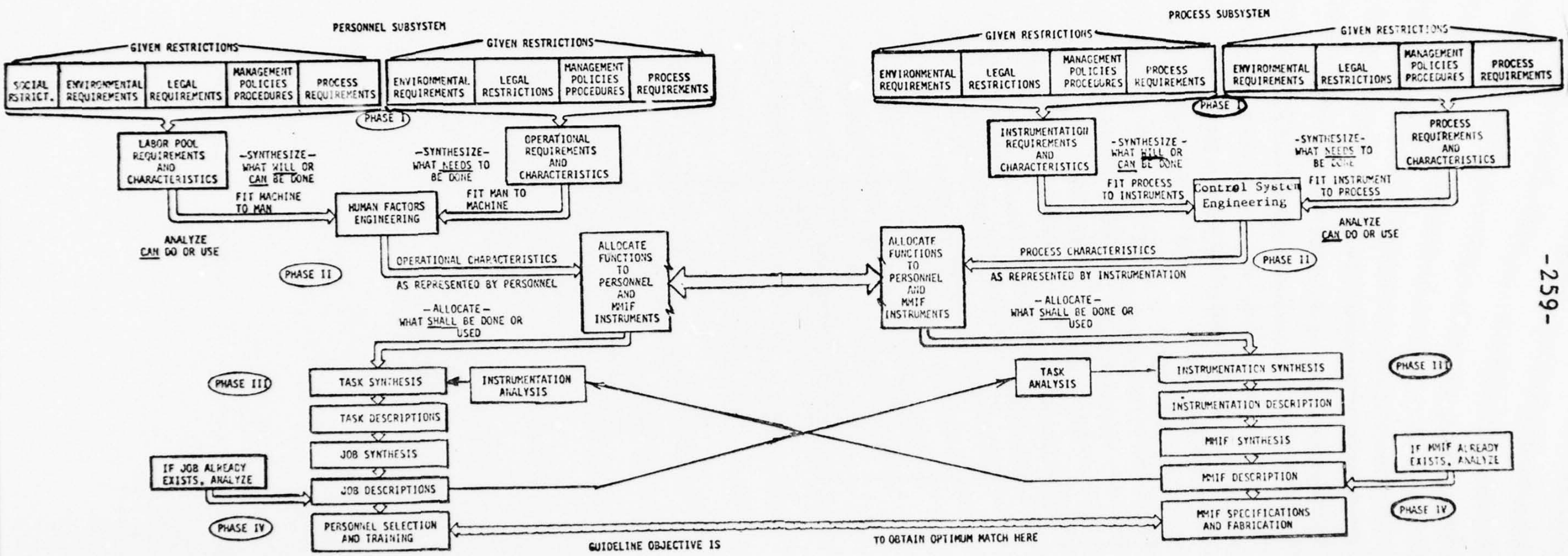


Fig. 3. The Guideline Flowchart of Man-Machine Systems Design.

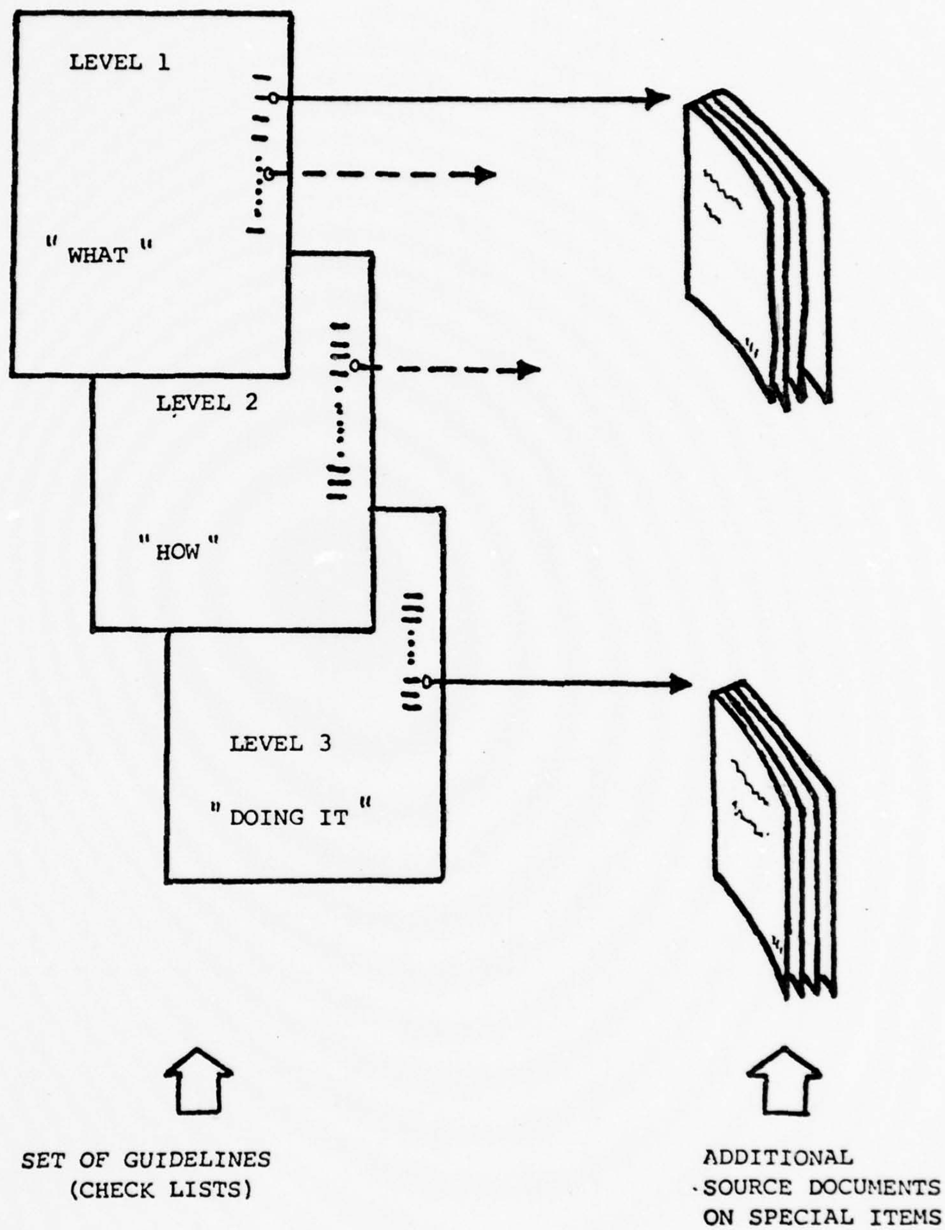


Fig. 4. Framework of a General Guideline Concept.

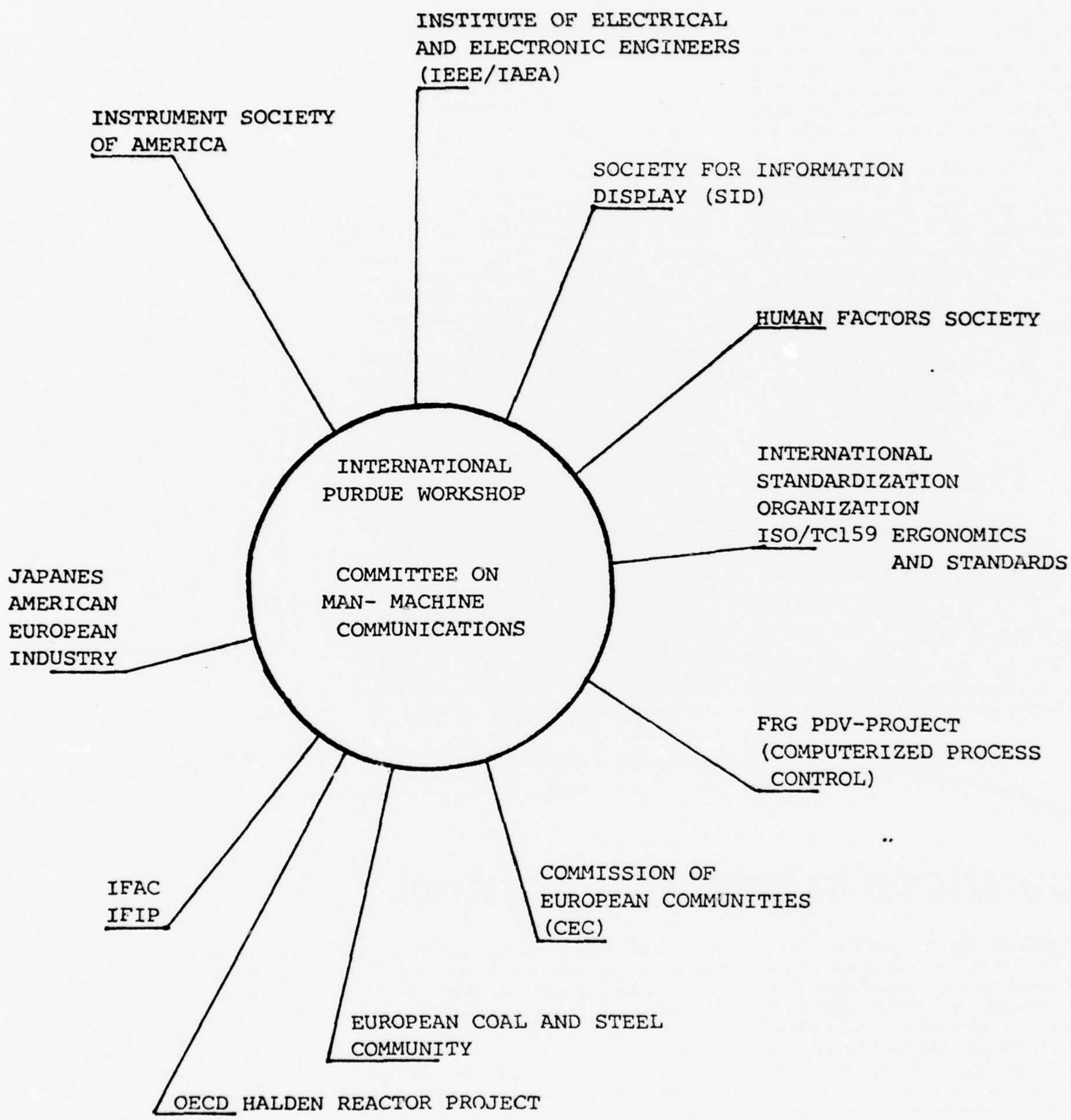
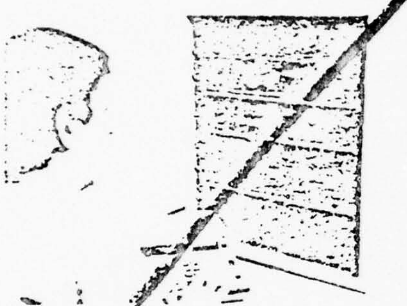


Fig. 5. Network of information SOURCES

wire resistance measurement, single channel monitoring and the ability to operate into a wide variety of output peripherals either singly or in a multiple mode. Signal measurement is by 4½ or 5½ digit premium digital voltmeters with 1μV resolution. Signal switching is by the 1200 Series Scanner/Clock which may have either conventional sequential scanning or computer control via the Bus. Systems on show will be: 60-channel conventional system with printer and tape cartridge, punch or teletype output; 30-channel IEC Bus system operating with Tektronix 4051 Graphic computer; and single-channel low cost computer controlled precision DVM.

Also featured will be the Model 1030 low frequency true RMS DVM, Ectron instrumentation amplifiers, thermocouple simulator/calibrator, and a high-speed amplifier per channel data acquisition system with up to 100 kHz throughput and 14-bit ADC.



Datron 1200 Series data logger

Jackson Transducer Systems (J4)

DC/DC capacitive displacement transducers. The standard range of angular devices gives effective linear electrical output over

The Kuflex multichannel P-I converter system provides a current or voltage signal from a standard 3–15 lb/in^2 pneumatic line for computer interface. Up to 21 P-I modules can be mounted on a standard 5¼ in high, 19 in rack assembly. Each channel has provision for output monitoring. Zero and span adjustment is provided. Both four-wire and two-wire versions are available. Inputs of 3–15, 3–27 and 20–30 lb/in^2 are accepted, with outputs of 0–10 V, 4–20 mA.

Prosser Scientific Instruments (K9)

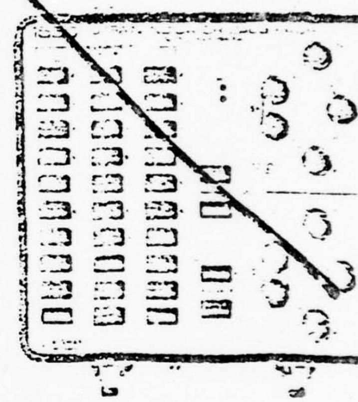
The new 6000 Constant Temperature Anemometer is a modular instrument for measuring flow, velocity, mean velocity, turbulence and percentage turbulence in air, gases and liquids. Plug-in modules available include anemometer bridge, automatic polynomial lineariser, manually-set lineariser, signal conditioners, analogue/digital meter unit, and power supply. It is compatible with all types of probes commercially available. The automatic lineariser module allows any probe to be used in its linear mode automatically without any need for lengthy calculations. Also showing will be a laser-doppler system which enables velocity of fluids and the speed of moving objects to be measured accurately without physical contact or disturbance, air velocity meters offer measurement facilities in the 0 to 30 m/s range, and thermal converters by the newly-formed Transducer Division at Felixstowe.

Pye Unicam (H2/H4)

The new Philips RMS700 range of equipment for the monitoring of rotational machinery. Other equipment to be shown includes the new eight-channel strain, torque and temperature measuring versions of the Philips FM system for short-range telemetry from

test dynamic signal conditioners/amplifiers. This is a DC amplifier system for strain gauges, transducers and other resistive devices. Band width is DC to 25 kHz with four built-in filters. The gain is variable from 1 to 11 000. Automatic bridge zero balance is a feature, and bridge supplies, independent for each channel, are variable from 0.5 to 15 V DC. Three simultaneous buffered outputs for oscilloscope, magnetic tape and galvanometer recorder are provided, and the filters can be used on playback from tape recordings. Internal calibration steps both ± are provided and the values are easily changed by the user.

The new Model 1550 strain indicator calibrator has true Wheatstone Bridge circuitry, and simulates quarter, half and full bridge, both 120 Ω/350 Ω. There are three decades of push-buttons: strain range direct reading: ±100 000 με in increments of 100 με; transducer range: ±50 mV/V in increments of 0.5 mV/V.



Welwyn 1550 strain indicator calibrator

Operators in process control

A report on *The Operator Interface in Process Control in European Industries* has been produced by the Purdue-Europe Technical Committee on Man–Machine Communications. It has analysed the responses to a questionnaire sent to a sample of process industries and associated organisations in several European countries in order to accumulate an overall picture of current control room design philosophy, practice and experience and to pinpoint the problem areas which are felt to be of concern – especially those reflecting common interests, attitudes and experience. In all, 32 responses were received to the questionnaire, which covered such aspects as task allocation between man and machine, use of displays, alarm treatment, fault handling, manning and the use of human factors techniques in control system design. Particular questions were aimed at identifying specific problems concerning the process operator such as information display, operator selection and training, operator–computer relations, human stress and error, handling of emergency situations, etc.

Rapid developments in technology are resulting in an increasing gap between the incorporation of (often unproved) advanced devices and systems and an adequate treatment of the associated human factors. Increased size and complexity of plant with stringent requirements for control of production tolerances and environmental effects together with the need for effective response to abnormal conditions places the process operator in a key position, and careful design of his total work situation including the man–machine interface is of crucial importance. However, conflicts can arise between a designer's natural tendency to assign as much responsibility and control to machines as is technologically feasible and the need for the operator to maintain an appreciation of the importance of his job and overall responsibilities.

Copies of the report and further information on the work of the Technical Committee can be obtained from: Dr A. B. Aune, SINTEF – Division of Automatic Control, N-7034 Trondheim NTH, Norway.

REASONS FOR IMPLEMENTING AND NOT IMPLEMENTING MODERN MAN-MACHINE INTERFACE FUNCTIONS

The following list describe positive and negative factors which will influence any decisions regarding implementation of modern man-machine interfact functions. The designer should carefully consider the items in these lists (as they apply to his system) and balance the positive and negative factors.

POSITIVE FACTORS

The list of reasons for implementing a modern man-machine interface is divided into two groups. The first group contains these reasons which are generally applicable to any MMIF upgrade or new design. Those in the second group will usually be positive factors, but must be evaluated in terms of the particular application.

1. GENERAL FACTORS

1. Improves Operator Efficiency

1.1 Better Process Representation

1.2 Less Operator Error (saves process downtime and improves product quality)

1.3 Data Presentation Capability (pattern recognition, operation by exception)

1.4 Display/Calculation of Derived Variables

1.5 Better Utilization of Computer Capabilities (all capabilities will not be utilized if the MMIF is difficult for the operator to interact with)

2. Reduces Overall Size of the Control Room.

3. Reduces Maintenance (fewer control and indicators).

4. Permits Easier Upgrade for Process Growth and Change (adaptability and flexibility).

5. Enhances Operator's Job Enrichment

6. Eases Operator Training (especially for new operators).

7. Permits Replacement of Equipment Which is no Longer in Production.

8. Improves Process Optimization

9. Provides Better Managerial Capability

10. Allows Implementation of Distributed Architecture

2. SPECIFIC FACTORS

1. Operations Manpower May be Reduced (greater number of loops or batch handled by each operator; powerful capabilities may permit upsets to be handled by a single operator).

2. Implementations Using Modern Hardware and Architectures may be more Reliable and have Higher Availability than Older Designs (independent, redundant, or distributed processing; fewer active and/or higher reliability parts).

3. Installed Cost of the MMIF may be Lower (cabling costs reduced on large systems).

4. Operator Morale may be Improved (utilization of state-of-the-art equipment; managing computer plant control).

5. An Upgrade to an Existing Control System may Require a Modern MMIF for Optimal Operation.

NEGATIVE FACTORS

The following list of negative factors must be analyzed by the design engineer for his MMIF system. These factors are usually quoted as reasons not to implement a modern MMIF. However, in many cases, the actual design is misunderstood, and these factors may not be valid. In addition, the severity of these factors will depend heavily on whether an existing MMIF is to be updated or a new control system is to be implemented (negative factors are generally more severy for MMIF upgrades).

1. Present personnel may not be capable of designing and implementing the MMIF for a new system. They may feel more secure with the existing design and implementation.

2. Proven technology can be used. This offers less risk and is more available and better understood.

3. Operators and maintenance people and practices may be different with a new design.

4. Design may be overkill for the application; too elegant and of little "real" use.

5. New MMIF may have negative effects on operating personnel and their jobs.

6. Design may increase the degree of abstraction of the process.

7. Reliability will be decreased due to use of common hardware.

8. Government regulations prohibit or greatly restrict new MMIF design (especially government standards/ucensing).

9. Unions/operators may be concerned about job dislocations.

10. Control room/process environment may prohibit a new design.

11. Costs of a new MMIF

11.1 May Include Additional Design Costs for Design Time, Technology, Development, Training, or Staffing

11.2 May Include Additional Maintenance Costs. These Costs will Normally be Lower However, Due to Better Packaging, Less different Types of Consoles, Hardware Commonality, On-Line Diagnostics.

11.3 May Include Additional Training Costs for Maintenance People and Operator Retraining.

11.4 May Include Additional Hardware and Software Costs for a more Complex CPU System and Higher Utilization of new Technology.

11.5 May Include Additional Installation Costs. These Costs will Normally be Lower However, Due to Better Electrical Interfaces (especially cabling) and Smaller Physical size.

11.6 Are Not Directly Related to Profits (the "business" of the corporation).

12. Modern MMIF design may require a more sophisticated and expensive computer control system than would otherwise be required.

VALVE OPERATIONS/MOTOR OPERATIONS

Command:	Open	Start
	Close	Stop
	%	Speed
	Stop	

Control Parameters: Address
Verification Feedback Needed
(Timeout)
(Rate)

Verification Feedback: OK - Operation Started

Error - No verification
Operation timeout
Control status error
Security violation

CHANGE SETPOINT

Command: Enter
Ramp

Control Parameters: Address
Verification Feedback Needed
Value
(± Increment per unit Time)

Verification Feedback: OK - Operation Started

Error - No Verfication
Control Status Error
Security Violation

MODIFY TUNING PARAMETERS

Command: LEAD
LAG
PID

Control Parameters: Address
Verification Feedback
Mode (C, A, M, CAS)
Value

Verification Feedback: OK - Operation Started

Error - No Verification
Control Status Error
Security Violation

APPENDIX E-XI

TC-8

REAL-TIME OPERATING SYSTEMS COMMITTEE
PURDUE EUROPE

1. Up to Date Report - January 20, 1978.

INTERNATIONAL PURDUE WORKSHOP ON INDUSTRIAL COMPUTER SYSTEMS

PURDUE EUROPE

Technical Committee on Operating Systems

Author: TC 8	OP/SYS I-1-6
Institution:	
Date: January 20, 1978	

Title: Up to Date Report

Abstract:

This report represents the conclusions reached to date by the Technical Committee No. 8 on Real Time Operating Systems of the PURDUE Europe Workshop. It is intended that this report be interpreted as a state of work paper for TC 8 and not as its final conclusions.

PRECEDING PAGE BLANK - NOT FILMED

TABLE OF CONTENTS PAGE

TABLE OF CONTENTS PAGE

1. SCOPE AND GOALS OF TC 8

The scope of TC 8 on Real-Time Operating Systems is to consider DEFINITION, CONSTRUCTION and USE of Real-Time Operating Systems (RTOS). The goal of TC 8 is to make the construction of RTOS's more efficient and economical, and to attain maximum reliability and portability through the development of guidelines and machine independent concepts.

These guidelines and concepts should be achieved through the interchange of IDEAS and INFORMATION, through the use of a COMMON TERMINOLOGY and finally through the definition of an OPEN-ENDED MODEL of a RTOS together with rules for expansion and reduction.

The promotion of this model should be achieved by involving RTOS-constructors in the work of TC 8, by imparting the results of TC 8 in teaching and education and finally by proposing these results to the INTERNATIONAL PURDUE WORKSHOP and consequently to the appropriate national and international organizations responsible for standardization.

2. PROCESSES, PROCESSORS AND THE NEED FOR PROCESSOR MULTIPLEXING

Most real-time systems can be considered to consist of a network of intercommunicating processes running on a set of one or more processors.

A PROCESS is defined as an action in which the operations are carried out one at a time.

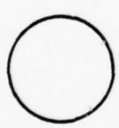
A PROCESSOR is capable of performing the operations (data manipulations) of a process (run a process). A processor can run at most one process at a time.

The maximum number of processes that can be physically run in parallel is equal to the number of processors.

Processors which are exchangeable with regard to the hardware configuration and the processes capable of being run by them can be treated as a PROCESSOR POOL and are dispatched with common dispatcher primitives. A processor pool with its set of exchangeable processors is identified by means of a PROCESSOR POOL DESCRIPTOR.

It is assumed that for each process which can run independently of others there exists a PSEUDO-PROCESSOR. All pseudo- processors have to be provided by the operating system. This requires the presence of some operating system primitives for processor multiplexing. The process is identified within the operating system by means of a PROCESS DESCRIPTOR.

It is convinient to represent a network of processes using the following diagrammatical notation:


to represent a process


to represent process-process communication


to represent a processor or a pool of identical processors

An example of a network described in this manner is (fig 1).

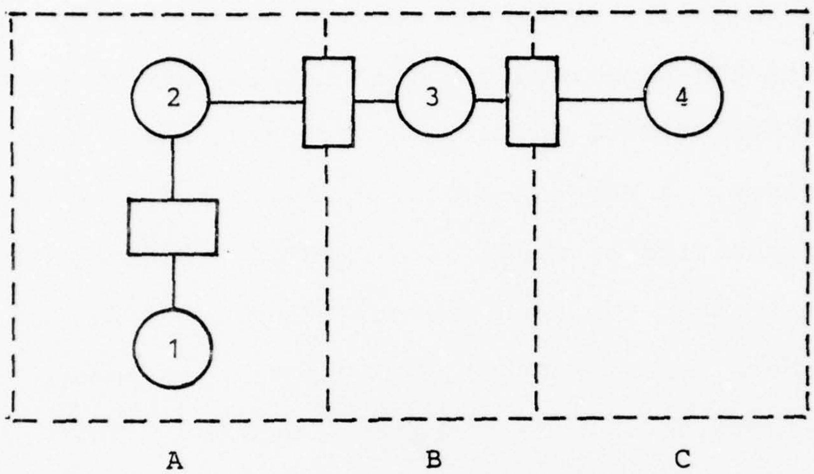


fig. 1

This depicts a network of four processes running on a three processor configuration.

3. STRUCTURE OF THE REAL-TIME OPERATING SYSTEM (RTOS)

There is a need to tailor the RTOS to specific applications. Tailoring is comprised of REDUCTION and EXTENSION of the basic RTOS. In particular this tailoring will include the provision of software interfaces to process peripherals. The fulfilment of this need for tailoring requires that the RTOS be highly structured and well engineered.

An approach embodying a hierarchical structure was adopted as a reasonable basis for the RTOS design and its logical management. The hierarchy is considered to consist of LEVELS OF CONSTRUCTION, where the realization of a new level from a lower level is achieved by a LAYER (fig. 2). The effect of each layer can be defined with respect to functions available at lower and upper boundaries.

This can be represented as follows:

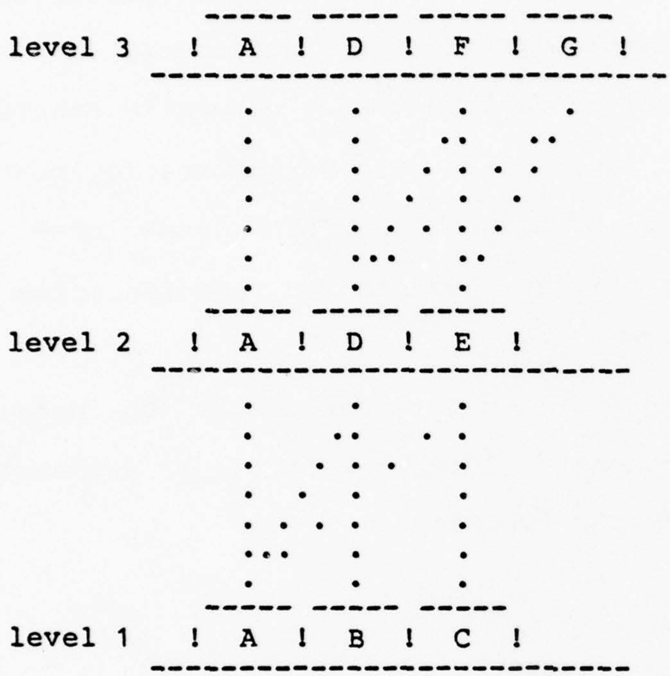


fig. 2

The row of connected rectangles lists the functions available at that interface level. A lower boundary function may be transmitted without any change to the upper boundary (e.g. A) or made completely invisible (e.g. B). Some new functions, not available at the lower level boundary, can be introduced by the layer (e.g. F). These new functions will be implemented within the layer making use of the functions available at the lower boundary. This is optionally indicated by lines connecting functions at adjacent levels.

The model for a RTOS to be defined is a logical model, i.e. it is to be implemented by an appropriate combination of hardware

and software.

The KERNEL is considered as the layer(s) which provides any number of pseudo-processors by muliplexing (physical) processors. The kernel is considered essentially strategy independent.

There exist on each level of implementation naturally embedded strategies. The choice of appropriate data structures for representation of processes as well as the sorting algorithms used in the ordering or selecting of list elements are representative of these strategies.

The higher levels of construction are necessary to provide both a convenient and comprehensive interface for 'user processes' and higher level operating system functions.

4. BASIC SYNCHRONIZATION

It is important to recognize that any synchronization mechanism is repeatedly built upon a more primitive mechanism, with the most primitive being hardware synchronisation. Thus the hardware has to provide facilities to implement primitives which exclude simultaneous access to common data.

The existence of two synchronisation primitives, LOCK and UNLOCK, is assumed. Both are considered indivisible operations and are not simultaneously executable by different processors on the same variable. They allow the realization of the BUSY-WAITING state of the physical processors.

The LOCK and UNLOCK primitives are used to bracket operations

which have to be executed mutually exclusively:

```
LOCK (v)
   ...
   mutually exclusive operations
   ...
UNLOCK (v)
```

The presence of the parameter 'v' (LOCKVARIABLE) shows that different groups of mutually exclusive operations exist; only those using the same lockvariable mutually exclude each other. 'v' has two possible states: 'LOCKED' and 'UNLOCKED'. The locked state represents the fact that a process is executing mutually exclusive operations and that no other process is allowed to "pass" through the LOCK primitive (with the same lockvariable). The UNLOCKED state is the reverse.

In systems with only one physical processor the LOCK and UNLOCK primitives can be realized with interrupt inhibition. Whether it is posssible to differentiate between different lockvariables depends on the existing hardware (selective interrupt inhibition). Structure of LOCK and UNLOCK for single (physical) processor systems:

```
LOCK (v)
inhibit interrupts (selectively corresponding to 'v').
```

UNLOCK (v)

enable interrupts (selectively corresponding to 'v').

In systems with more than one physical processor a busy-wait loop for processors waiting for the lockvariable to be unlocked has to be provided. In addition, interrupts still have to be inhibited for two reasons:

- The time between the execution of the LOCK and the UNLOCK primitives has to be minimized, because during this time other processors may stay in busy-wait loops. Interrupts being serviced between LOCK and UNLOCK extend this time and should therefore be inhibited, if possible.
- A deadlock situation arises when a processor stays in a loop waiting for a lockvariable which has already been locked by the same processor. Such a situation is possible when the processor has serviced an interrupt between the execution of the LOCK and UNLOCK primitives. The corresponding interrupts must therefore be inhibited between LOCK and UNLOCK.

In multiprocessor systems the two primitives have the following general structure:

LOCK (v)

switches the processor executing this function into a non-interruptable mode and then checks whether the lockvariable 'v' is already in the locked state. If not, 'v' is locked and execution of the mutually exclusive operations is started. If so, the processor is switched back into the mode it was previously in and

the function is restarted at its beginning, thus leading to a busy-wait loop until the lockvariable is switched to the unlocked state by another process. This other process is presently executing mutually exclusive operations locked by the same lockvariable.

It is important that checking 'v' and putting it to the locked state is executed as an indivisible operation which itself must be mutually exclusive with other possible accesses to 'v'. This facility must be basically supported by the processor-hardware.

UNLOCK (v)

switches 'v' to the unlocked state and switches the processor back to the mode it was in before it executed the corresponding LOCK primitive.

5. BASIC SUPPORT FUNCTIONS

5.1 LIST-MANAGEMENT

A list is a possibly empty collection of elements, which share some common property. Operations on lists may take account of some ordering strategy.

Note: Lists may be implemented using
- linked lists in the conventional sense,
- fixed length arrays of elements,
- subsets of more general lists, with bits set to indicate membership of a particular list.

Within all levels of the RTOS kernel list management functions

are necessary. They are defined as follows:

INSERT (el,list)

inserts the element 'el' into the list 'list' according to some strategy.

REMOVE (el,list)

selects an element from the list 'list' according to some strategy and returns it in the parameter 'el'. The element is removed from the list.

LISTSIZE (list,lenght)

the actual number of listelements within the list 'list' is returned in the parameter 'length'.

NEXT (el,list)

based on the possibly implementation-dependant linear ordering of the list 'list' (e.g. increasing addresses, index of arrays, link chain, sequence according to some strategy) the function NEXT returns in the transient parameter 'el' the successor to the input element 'el' of the list 'list'. The element is not removed from the list. The first element of the list is obtained if input to 'el' is empty, i.e. the parameter 'el' contains a predefined empty element symbol (NIL). If the list is empty or no more successor exists the output to 'el' is the empty element symbol.

SEARCHFOR (el,list,id_attribute)

removes one element of the list 'list', which has the identification attribute 'id_attribute' and returns it to the parameter

'el'. If there is more than one element with such an attribute the selection is strategy-dependent. The attribute consists of at least a pair of parameters: The kind of attribute (perhaps denoted by an index within the listelement) and its special value, e.g. ('process',task1), ('priority',10), ('key',123). Output to 'el' is empty (NIL), if the list contains no element with the specified attribute.

CHANGE (list, id_attribute, change_attribute)

the list 'list' is searched for all elements with the identification attribute 'id_attribute'. Within these elements the value of the attribute specified by 'change_attribute' is reassigned. 'change_attribute' is of the same type as 'id_attribute' and contains both kind of attribute, which has to be changed, and the new value of that attribute.

5.2 Context Switching

The CONTEXT of a process is that part of the information belonging to a process, which, when the process is actually executed by a processor (process is in the RUNNING state, cf. par. 6), is located in processor-specific storage (locations and/or registers). This storage is used by all processes running on that processor. Although size and structure of a context may vary in a wide range, a context exists in every case.

That part of the context which is used by the kernel must be

saved on entering the kernel and is restored on leaving the kernel; some of these operations are performed by hardware (e.g. saving and restoring of program counter, processor status).

If, within the kernel, processes have to be switched, two functions operating on contexts are necessary:

SAVE_CONTEXT (process)

moves the context of 'process' into the process descriptor of 'process'.

RESTORE_CONTEXT (process)

restores the context of 'process' out of the process descriptor in a way that will enable the process to be continued on exit from the kernel. The initial state of the context of a process stored in its descriptor is defined such that the function RESTORE CONTEXT will enable the process to start correctly.

6. PROCESS- AND PROCESSOR-STATES

Within a multiprocessor kernel process- and processor- states have to be considered. The dispatcher primitives will cover all necessary state-transitions for processes and processors (fig. 3)

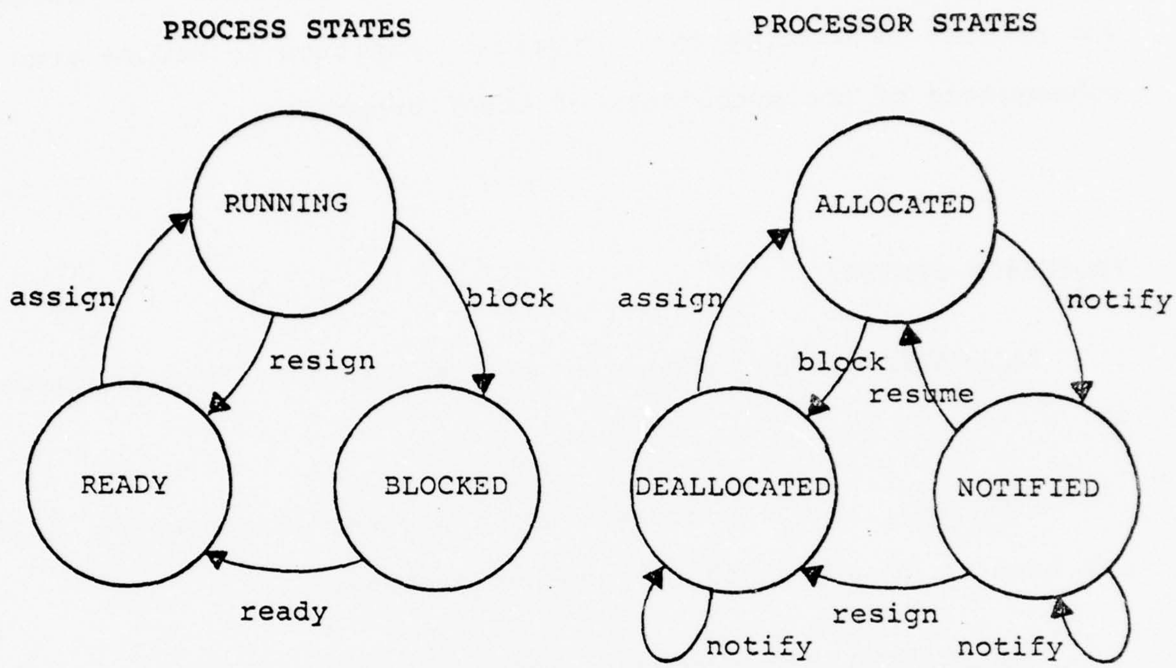


fig. 3

The ASSIGN, RESIGN and BLOCK primitives apply to both processes and processors. The READY primitive is unique to processes, whilst the NOTIFY and RESUME primitives are unique to processors.

PROCESS STATES:

RUNNING......The process has a suitable processor allocated to it.

READY........The process is competing for a specific processor or one of a pool of processors.

BLOCKED......The process is not competing for a processor. The process is waiting for a specific condition to become true as a consequence of the operations of other processes.

PROCESSOR STATES:

ALLOCATED....The processor has a suitable process allocated to it.

NOTIFIED.....The processor has a process allocated to it which may have to be exchanged for another process.

DEALLOCATED..The processor has no process allocated to it.

DISPATCHER LISTS

We assume the presence of different lists which are accessed and/or modified by the dispatcher primitives.

- Each process descriptor contains the following elements:
  - identification of the processor-pool which is considered to execute the corresponding process (this information may be implicite).
  - storage to save its context

- Each processor pool descriptor contains the following elements:
  - the identification of the RUNNING process (if any) for each processor of the pool
  - a list of all processes which are READY to run on that pool (READYLIST).

Note: Asynchronous program executions must exclude each other in time while accessing shared data. Regarding the data accessed by the dispatcher primitives, mutual exclusion is achieved using the LOCK and UNLOCK primitives.

DISPATCHER PRIMITIVES

ASSIGN

Removes a process (if there is any) from the readylist of the pool of the processor executing the primitive, assigns it to this processor, switches the removed process from READY to RUNNING and the executing processor from DEALLOCATED to ALLOCATED. It restores the context of the removed process calling RESTORE_CONTEXT.

If there is no READY process in the corresponding readylist, the processor remains in the DEALLOCATED state and waits.

Note: If 'idle' processes exist, there is always a READY process in the readylist and the processor will not remain in the DEALLOCATED state. If no 'idle' process exists, the REDISPATCH function (cf. par. 7) will cause the processor to leave the DEALLOCATED state. It is implementation dependent whether a processor waiting in the DEALLOCATED state should periodically check for the presence of READY processes. If not, the NOTIFY and REDISPATCH functions are mandatory, if yes, they may be omitted in non-preemptive systems (cf. par. 7).

RESIGN

Decouples the process from the processor executing the primitive and saves its context calling SAVE_CONTEXT. Inserts the process in the readylist of the corresponding processor- pool, switches the process from RUNNING to READY and the executing processor from NOTIFIED to DEALLOCATED. An embedded strategy for the ordering of the READY processes in the readylist is used by the called INSERT function.

BLOCK

Decouples the process executing the primitive from its allocated processor and saves its context calling SAVE_CONTEXT. Switches the process from RUNNING to BLOCKED and the processor from ALLOCATED to DEALLOCATED.

READY (process)

Switches 'process' from BLOCKED to READY and inserts it into the readylist of the corresponding processor-pool. An embedded strategy for the ordering of the READY processes in the readylist is used by the called INSERT function.

NOTIFY (processor-pool)

Switches a processor of the 'processor-pool' from ALLOCATED to NOTIFIED by sending a stimulus. If the processor receiving the stimulus is in the DEALLOCATED or NOTIFIED state, no change of state occurs. If the pool consists of more than one processor, selection of the processor receiving the stimulus is subject to some embedded strategy. (It is assumed that processors in the DEALLOCATED state, if there are any, are selected first.)

As a result of receiving the stimulus the processor will execute the REDISPATCH function (see par. 7).

Note: The reason for using a stimulus rather than a call to start the REDISPATCH function is that the sending and the receiving processor in general are not identical and executing completely asynchronously at the time of the notification. If the sending and receiving processors are identical, the NOTIFY primitive can lead to a direct call of the REDISPATCH function.

RESUME

Switches the processor executing the primitive from NOTIFIED to ALLOCATED. Thus, the processor will continue to execute its RUNNING process after leaving the kernel.

7. PREEMPTION AND PROCESSOR REDISPATCHING

Separation of a RUNNING process from its processor (executing the RESIGN primitive) in order to free this processor to ASSIGN another, presently more important process, is called preemption.

As opposed to the RESIGN primitive the BLOCK primitive is executed by a process which voluntarily stops execution and thus frees a processor.

Preemption is not necessarily a property of a RTOS. If there are either always enough processors to execute the necessary processes, or if the different competing processes are known to always execute the BLOCK primitive in time to allow the other processes to run correctly, a simpler model of process- and processor states and state transitions is possible (see fig. 4).

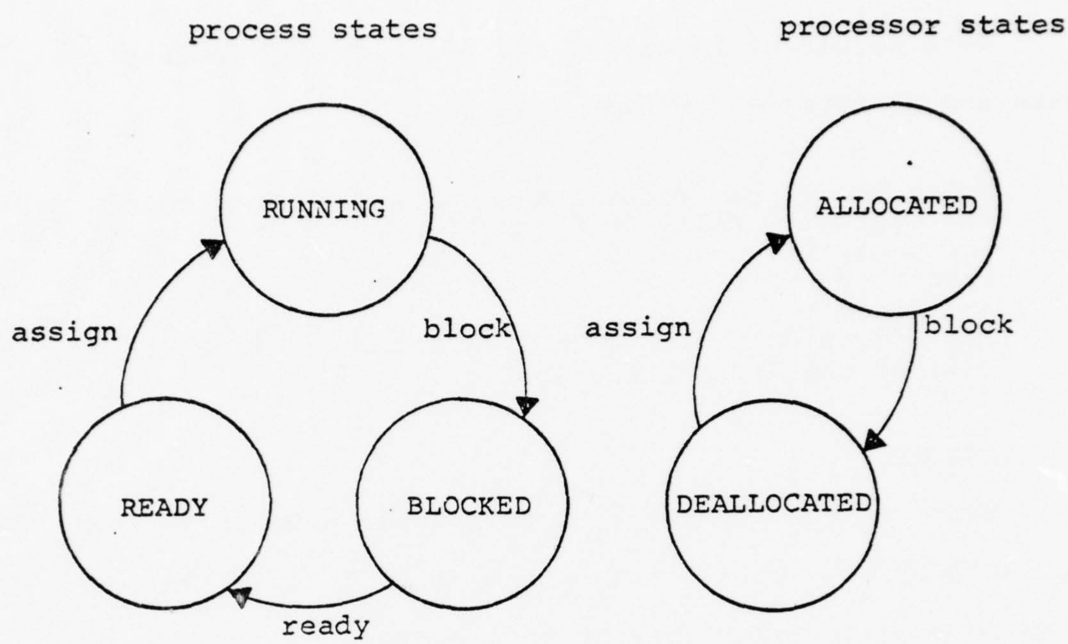


fig. 4

The RESIGN, NOTIFY and RESUME primitives and the NOTIFIED state are not contained in a system without preemption.

Note: A system with 'idle' processes will in most cases need preemption.

In systems with preemption a preemption may become necessary whenever a READY process has been added to a readylist or, more general, when changes have been made to the readylist. If after a change to the readylist a (possible) preemption is desired, the process which made the change has to NOTIFY the corresponding processor- pool. A processor of this pool will then execute the REDISPATCH function:

REDISPATCH

The processor executing this function is either in the NOTIFIED or in the DEALLOCATED state. In the DEALLOCATED state, it will execute the ASSIGN primitive and thus start execution of a previously READIED process, if there was any, or stay in the DEALLOCATED state. In the NOTIFIED state, it will check whether it is more important to RESUME its RUNNING process or to RESIGN this process and to ASSIGN a new one.

Note 1: The detailed construction of the REDISPATCH function is implementation-dependent. The RESUME function can be avoided, if the NOTIFY primitive is only executed when the conditions which lead to preemption are true. It also can be avoided, if the REDISPATCH function unconditionally RESIGNS the RUNNING process, even if this same process is then ASSIGNed again.

Note 2: The voluntary releasing of a processor by a process in favour of other processes is considered a preemption. Accordingly, such processes will execute the NOTIFY primitive which triggers the REDISPATCH function. Since in this case the sending and receiving processors of the NOTIFY-stimulus are always identical, simple implementations are possible (see par. 6, description of the NOTIFY primitive).

8. PROCESS MANAGEMENT

It is necessary to have functions for the CREATION and DELETION of processes and for the STARTING and TERMINATING/ RETIRING of processes. These functions also introduce two new process states: UNDEFINED and INACTIVE. The new functions and states are expressed in the following state transition diagram together with existing functions and states (fig. 5):

Note that only the terminal nodes of the tree are actual process states. All other states are for convenience of description only.

The new process states are:

UNDEFINED....An undefined process is non-existent.

INACTIVE.....The process and its process descriptor have been created but the process is inactive and not subject to control by the dispatcher.

The process management functions are:

CREATE

Takes a set of system elements (assumed to be 'building blocks'. created by previous stages of software construction, e.g. compiler, linker, etc.) and builds a process and its process descriptor. The process thus constructed is put into the INACTIVE state.

START (process)

Switches the process from the INACTIVE to the READY state with appropriate initialization (e.g. process descriptor, process parameters, etc.). The process thus becomes ACTIVE and RUNNABLE.

RETIRE

The RUNNING process voluntarily retires from further dispatcher influence and is placed in the INACTIVE state.

TERMINATE (process)

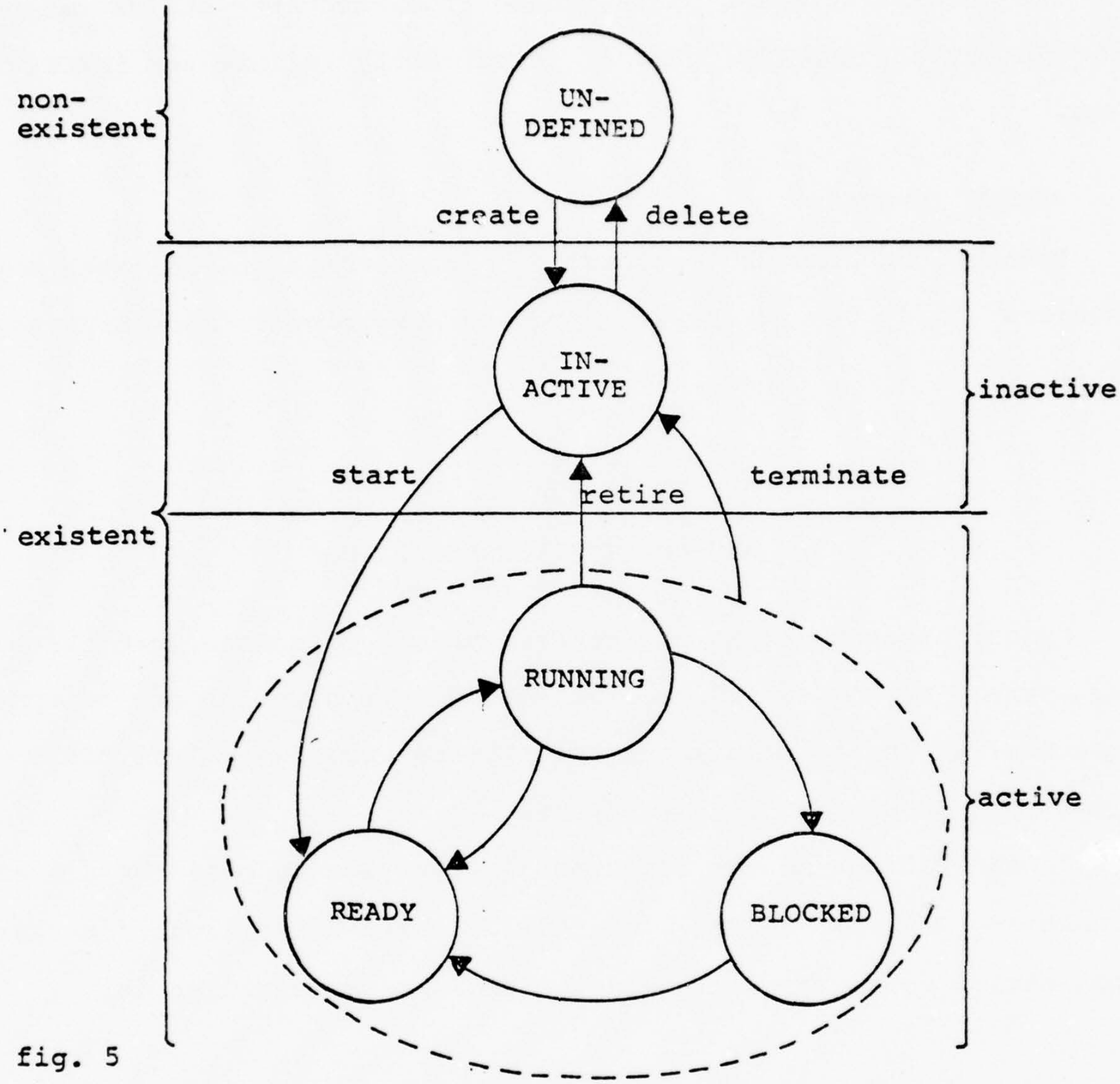


fig. 5

It is useful to indicate a hierarchy of conceptual states in a tree structure (fig 6):

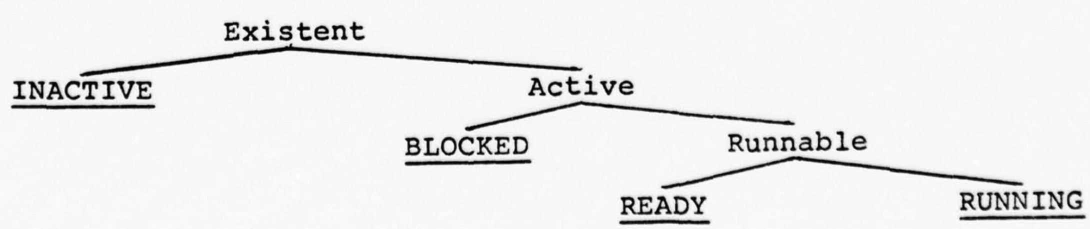


FIG. 6

The process 'process' is switched from whichever ACTIVE state (RUNNING, READY, BLOCKED) that it is currently in into the INACTIVE state.

DELETE (process)

Removes the process 'process' from the system and releases any resources held by it (e.g. process descriptor, main storage, etc.).

9. SYNCHRONIZATION FUNCTIONS

Synchronization tools are necessary to coordinate processes. The kernel is now suited to implement a more powerful set of synchronization functions in which the physical processors are disconnected from processes going to 'wait'.

Considerations of the fundamental requirements posed by a synchronizing concept result in the following definition of a synchronization model (fig. 7) and two actions performed on it.

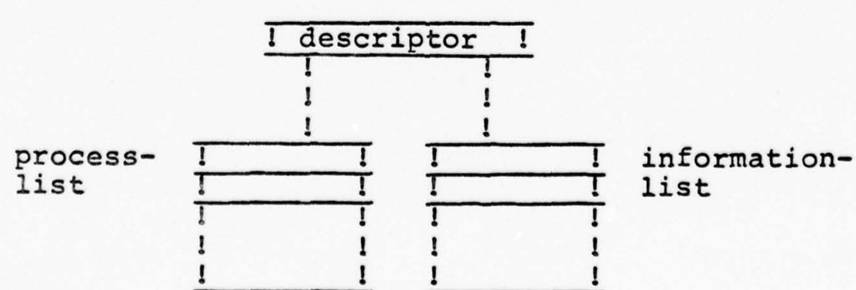


fig. 7

SEND-ACTION (information)

process list = (empty)	insert information into information list
= (not empty)	remove at least one process from process list, deliver information to it and READY it

WAIT-ACTION (information)

information list = (empty)	insert executing process into process list, BLOCK executing process
= (not empty)	remove at least one *element* from information list, deliver it to executing process

From the definition of these two actions it is obvious that only one of the two lists is present at a time and that both lists have identical elements (process identification and information as minimal set). Therefore we only need space for one list and only one set of list functions to implement the synchronization model. To distinguish the two lists a counter is introduced into the descriptor with the following meening:

counter = 0 :	information list = (empty) AND process list = (empty)

counter > 0 : information list = (not empty) AND
process list = (empty)

counter < 0 : information list = (not empty) AND
information list = (empty)

abs(counter) = number of elements in either list.

These relations assume that the corresponding lists exist. It should be mentioned that all strategy-dependent parts of the synchronization functions are embedded into the list functions.

The counter and the corresponding list form an entity called a SYNCHRONIZATION ELEMENT.

To avoid any confusion with established synchronization concepts, the two functions are now called INC (send- action) and DEC (wait- action), reflecting their operation on the counter.

At least two data elements are accessed by the two functions: The synchronization element itself and the readylist of a processor-pool. Simultaneous access to these elements has to be excluded using the LOCK/ UNLOCK functions. The two functions are described as follows:

INC (synchelement, info)

The counter of the synchronization element 'synchelement' is incremented. If the counter is now less than or equal to zero, at least one process is waiting in the process list. One process is removed, switched to the READY state and the information 'info' is passed to the place formerly defined when the now removed process

executed the corresponding DEC function. The processor- pool of the removed process is (conditionally) NOTIFIED (cf. par. 7). If the counter is greater than zero, no process is waiting and the (identification of the) sending process and the information 'info' are inserted in the information list.

DEC (synchelement, info)

The counter of the synchronization element 'synchelement' is decremented. If the counter is now less than zero, no information is available. The process executing the function and a pointer to 'info' (the place where the information will have to be put) is inserted into the process list and this process is put in the BLOCKED state. The executing processor becomes free and executes the ASSIGN function.

If the counter is equal to or greater than zero, the necessary information is removed from the information list and passed to the return parameter 'info'.

10. REALIZATION OF SOME HIGHER LEVEL SYNCHRONIZATION CONCEPTS

Higher level synchronization functions can be realized using INC and DEC. Usually, the interface to the higher level synchronization tools constructed in this layer is built by SVC (supervisor call) or TRAP instructions.

The following examples of some known synchronizing concepts illustrate the method.

10.1 SEMAPHORE

The semaphore concept introduced by E.W. Dijkstra can be realized easily.

For each semaphore one synchelement sys_sema must be initialized. (Initial value of sys_sema.counter >= 0)

Implementation of the two operations on semaphores:

```
V(sema) = INC (sys_sema,NIL);
P(sema) = DEC (sys_sema,NIL);
```

The use of NIL implies that the information-queue does not exist and all operations which are performed on it within INC and DEC may be omitted (to speed up operation).

10.2 MESSAGE

For each process one synchelement sym_proc must exist (creation static or dynamic).

Buffer allocation under process control or RTOS control for internal use of messages.

```
SENDM(proc,message) = INC(sym_proc,message);
WAITM(message)      = DEC(sym_actproc,message);
```

10.3 MONITOR

Monitors are realized based on the definition of C.A.R. Hoare.

A monitor is a collection of associated data and procedures working on this data. Processes may at any time attempt to call such a monitor procedure; however only one process at a time succeeds in entering it. Signal and wait operations on condition variables are provided within the monitor to delay a process. When a process signals a condition it must wait until the resumed process (if there is one) permits it to proceed.

For each monitor the following synchelements are necessary:

symon_monid	-	to realize exclusive entry into the monitor
syurgent_monid	-	to notify which signalling processes are waiting
sycond_cvar	-	one for each condition variable

monid = identifier for monitor
cvar = identifier for condition variable

Translation of monitor syntax into supervisor calls of synchronization functions can be expressed as follows:

```
                         ENTER (monid);
monid.procedurename  ->  procedurename;
                         EXIT (monid);

cvar.signal          ->  SIGNAL (cvar,monid);
cvar.wait            ->  WAIT (cvar,monid);
```

Implementation of these synchronization functions by the elementary synchronization functions INC and DEC:

```
ENTER (monid)        = DEC (symon_monid,NIL);
EXIT (monid)         = IF syurgent_monid.counter < 0
                       THEN INC (syurgent_monid,NIL)
                       ELSE INC (symon_monid,NIL);
WAIT (cvar,monid)    = EXIT (monid);
                       DEC (sycond_cvar,NIL);
SIGNAL (cvar,monid) = IF sycond_cvar.conter < 0 THEN
                       BEGIN INC (sycond_cvar,NIL);
                             DEC (syurgent_monid,NIL);
                       END;
```

If every SIGNAL occurs as the last statement of its monitor procedure, the synchelement syurgent_monid can be omitted together with all operations on it.

```
ENTER (monid)        = DEC (symon_monid,NIL);
EXIT (monid)         = INC (symon_monid,NIL);
WAIT (cvar,monid)    = INC (symon_monid,NIL);
                       DEC (sycond_cvar,NIL);
SIEXIT (cvar,monid) = IF sycond_cvar.conter < 0
                       THEN INC (sycond_cvar,NIL)
                       ELSE INC (symon_monid,NIL);
```

More complex monitors may be implemented if WAIT and SIGNAL are replaced by message functions.

11. INPUT / OUTPUT

Input/Output is defined to be the movement of data into or out of the system and can be viewed as an extension of inter-process communication.

Input/Output processing is considered to consist of two cooperating parallel processes, one process running in a general purpose processor and the other process running in a conceptually separate I/O processor. Depending upon the hardware configuration, the I/O process may be executed on a seperate physical processor or alternatively, the logic of the I/O process may actually be executed by a general purpose processor.

The configuration of two physically separate processors is represented in fig. 8

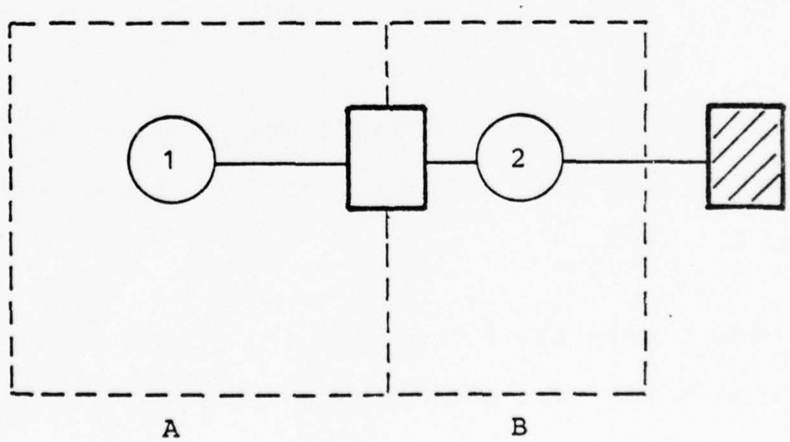


fig. 8

where  denotes hardware source/sink

In this figure process 2 is the I/O process running on the physically separate I/O processor B.

The configuration of two conceptually separate processors, realized by only one physical processor, is represented in fig. 9.

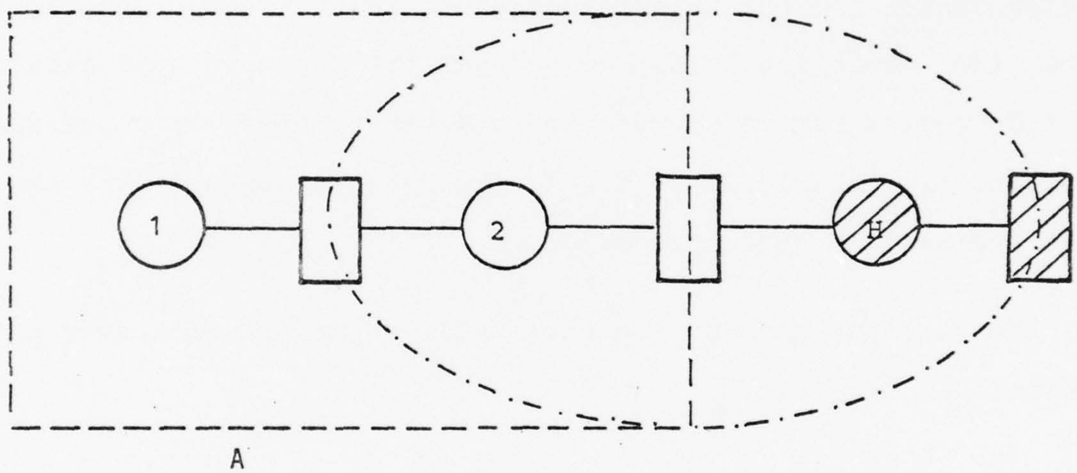


fig. 9

denotes the hardware logic which is conceptually a process communicating between hardware source/sink and the I/O port of the processor A.

In this figure the conceptual I/O processor is represented by the dash-dot line. Two cooperating processes run on this conceptual processor: Process 2 - a software I/O process actually running on the physical processor A and process H - a hardware process.

Note that process 2 is not scheduled by the dispatcher but is scheduled by the hardware interrupt which thus provides an alternative and more primitive method of processor multiplexing.

A more conventional view of this configuration is shown in fig. 10.

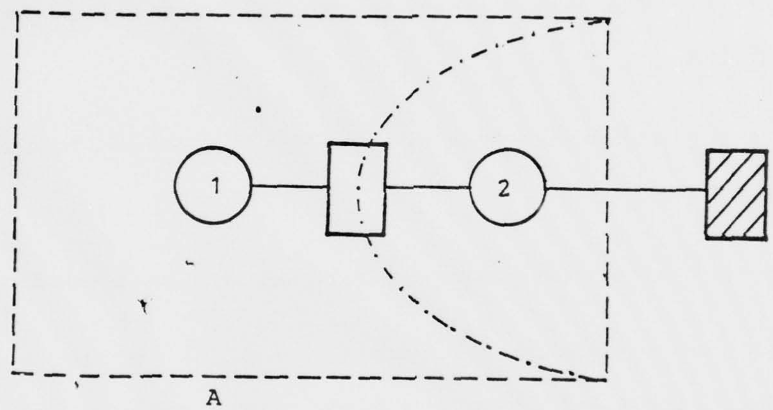


fig. 10

The hardware source/sink now includes the hardware process H and the I/O communication port of the processor A.

12. EXCEPTION HANDLING

Four types of exception handling have been identified:

(a) exceptions detected in function calls
(b) exceptions (excluding(a)) attributable to the running process (e.g. arithmetic overflow, store violation)

(c) exceptions not attributable to the running process and detected by the operating system (e.g. store parity failure)

(d) exceptions detected by one process and attributable to another process

Type (a) exceptions are best handled by a return parameter of each function.

Type (b) exceptions can optionally be handled by a user defined routine.

Type (c) exceptions must be handled by the operating system.

Note that exceptions of type (b) and (c) occur synchronously as a direct result of executing a processor instruction.

Type (d) exceptions are the responsibility of the application system and are not to be handled by the operating system.

It is not appropriate to state the action to be taken upon the occurrence of an exception as part of a set of guidelines or standards. However, the guidelines should include some mechanism by which the exceptions may interface with the process above the kernel.

One internal function is introduced to support the handling of type (b) and (c) exceptions:

EXCEPTION_RECEIPT

The function receives the exception upon its ocurrence. It must determine whether the exception is of type (b) or (c). If it is of type (b) and a user defined routine has been specified, control is transferred to that routine after appropriate manipulation of process descriptor, stack(s) etc. For type (b) and (c), when no user defined routine has been specified, control is transferred to an operating system exception handling process.

To enable the process to control the setting of the exception handling routine two functions are introduced:

EXCEPTION_ON (address of user defined routine)

sets the exception handling routine for type (b) exceptions to a user defined routine identified in the parameter.

EXCEPTION_OFF

sets the exception handling routine for type (b) exceptions to the default option provided by the operating system.

One important decision has to be taken by a user defined exception handling routine, that is to decide whether the process can continue or whether it must be stopped. In the latter case the process management function RETIRE will be called. For the former case the following function is introduced:

EXCEPTION_RETURN

Returns control back to the operating system to enable it to do any necessary manipulations of process descriptor, stacks, etc. before returning control to the exception producing process.

13. DEFINITION OF AN OPERATING SYSTEM KERNEL

According to the model presented in fig. 2 the proposed operating system functions and their hierarchical relations are shown in fig. 11

The first three levels of the system can be regarded as a KERNEL system. Two types of entries to that kernel can be distinguished:

synchronous or procedure-like entries (called by a process):

INC, DEC, NOTIFY

EXCEPTION_ON, EXCEPTION_OFF

asynchronous or interrupt-like entries:

REDISPATCH

EXCEPTION_RECEIPT

NOTES:

- "EXCEPTION HANDLING" on level 2 represents some necessary, but not yet defined functions to handle exceptions;
- the process management functions will also need additional functions within the kernel and are not included in the figure;
- other functions to access kernel data, to reorder lists, etc., will have to be included in the kernel;
- the various relations of the LOCK and UNLOCK functions are not explicitly shown in the figure.

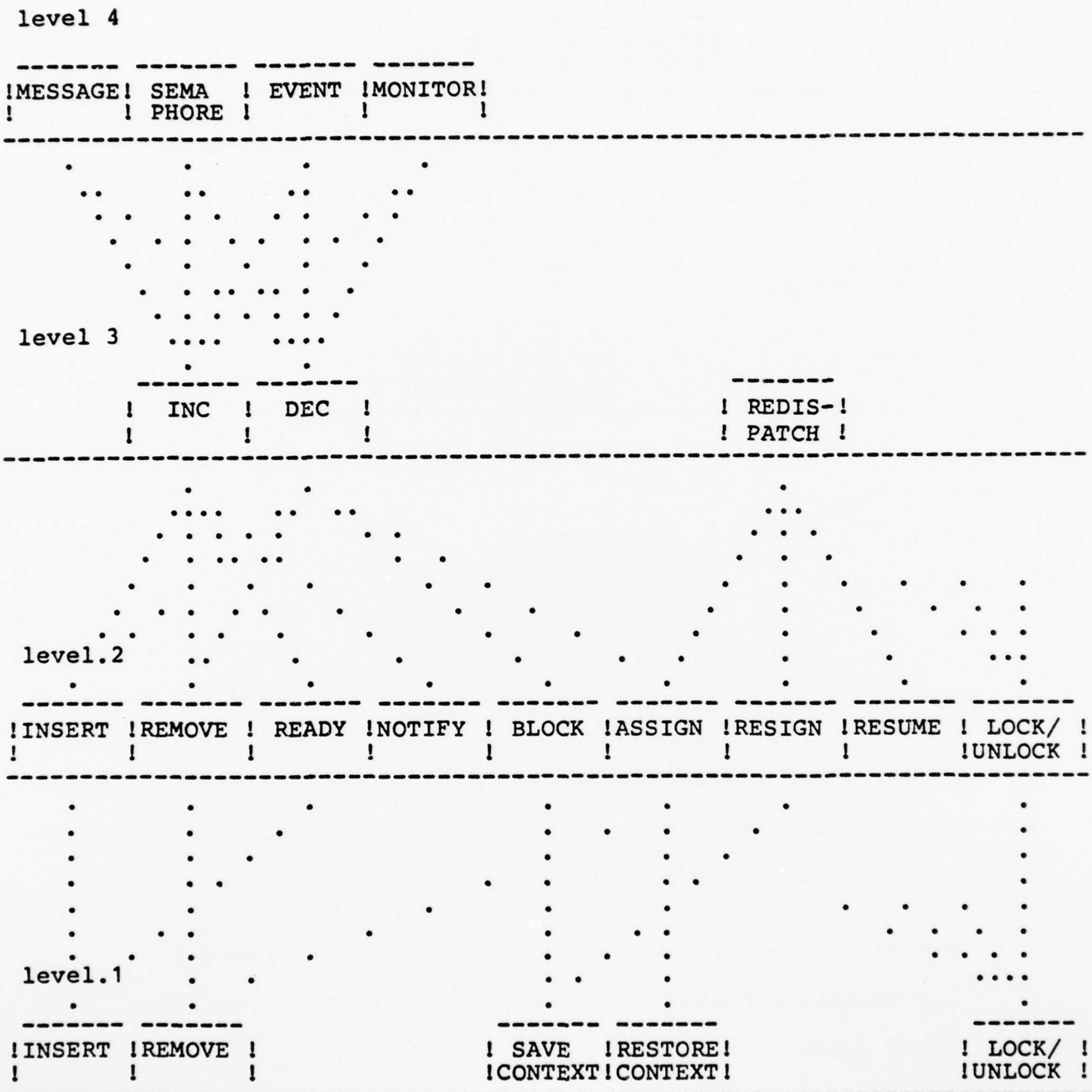

level 4
MESSAGE
SEMA PHORE
EVENT
MONITOR
level 3
INC
DEC
REDIS- PATCH
level.2
INSERT
REMOVE
READY
NOTIFY
BLOCK
ASSIGN
RESIGN
RESUME
LOCK/ UNLOCK
level.1
INSERT
REMOVE
SAVE CONTEXT
RESTORE CONTEXT
LOCK/ UNLOCK

fig 11a

```
level 3

           ------- ------- -------
          !EXCEPT.!EXCEPT.!EXCEPT.!
          !RECEIPT!  ON   !  OFF  !
  -----------------------------------------
              .       .       .
               .      .      .
                .     .     .
                 .    .    .
                  .   .   .
                   .  .  .
                    . . .
level 2              ...
                      .
                  ------- -------
                 !EXCEPT.!EXCEPT.!
                 !HANDL. !RETURN !
  -----------------------------------------
```

fig. 11b

14. FUTURE WORK

Chapters 1 to 13 report the currently agreed specifications for the design of the operating system. The objectives of TC 8 for the next year are to complete the kernel and to continue work on higher levels of construction.

Outstanding work on the kernel includes the following:

- the process management and exception handling functions have to be fully specified.
- kernel data management - this covers all the imbedded strategies including the ordering and reordering of lists etc. It also includes strategies for selective lock-outs on data areas which avoid deadly embrace situations.

- system configuration and reconfiguration - both in response to manual requests and as a consequence of exception recovery.

Among the topics to be tackled in the layers above the kernel are:

- resource allocation
- store management

It is likely that this program of work will have repercussions on existing specifications and therefore a further iteration of these can be expected.

APPENDIX A OF TC 8 REPORT

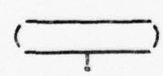
NOTATION OF THE OPERATING SYSTEM FUNCTIONS

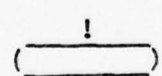
This appendix shows in a more detailed notation the construction of the described operating system functions. As opposed to the general descriptions in the report, the detailed notations may contain implementation dependent parts.

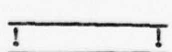
A PASCAL-like notation and/or flowcharts are used to describe the functions. The numbering of the paragraphs corresponds to the first part of the report.

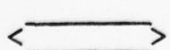
Comments on the PASCAL-like notation:
Not all data elements used are described completely; the data descriptors are such that the functions described are understood without unnecessarily prescribing a specific implementation.

Comments on the flowcharts:
The following graphical elements are used:


synchronous entry


synchronous exit


operation(s)


conditional branches

```
----
    /
    ---->(________)          asynchronous entry
             !
```

```
        !
(_________)----              asynchronous exit
              /
              ---->
```

A.4. Basic Synchronization

```
                           ( LOCK  )      parameter: v
                               !
                               !---------<-------------
                               !                      !
                      ! inhibit interrupts !          !
                      !   (*selectively*)  !          !
                               !                      !
                               !          no          !
           indivisible (  < v=unlocked  >-->--! enable interrupts !
           operation   (    ! v:=locked !     !   (*selectively*)  !
                               !
                               !
                            (______)
```

```
                          ( UNLOCK  )      parameter: v
                              !
                              !
                        ! v:=unlocked !
                              !
                              !
                     ! enable interrupts !
                     !   (*selectively*)  !
                              !
                              !
                           (______)
```

Pascal notation:

```
TYPE protection = (locked,unlocked);

PROCEDURE LOCK (VAR v: protection);
   BEGIN inhibit
      WHILE v = locked DO
         BEGIN
         enable; inhibit
         END;
      v:=locked
   END;


PROCEDURE UNLOCK (VAR v: protection);
   BEGIN
      v:=unlocked
      enable
   END;
```

where enable - procedure to enable interrupts
inhibit - procedure to inhibit interrupts

A.6. Process- and Processor States

```
                          ( ASSIGN  )
                              !
                              !- - - - -<- - - - -
                              !          no      !
             < READY process exists  >->-! wait  !
                              !
                              !
       !     LOCK (lockvariable of pool)        !
       ! (*pool of the executing processor*)    !
                              !
                              !
      !     REMOVE (newproc, readylist)          !
      ! (*readylist of executing processor*)     !
                              !
                              !
         !   state of newproc:= RUNNING        !
         ! state of processor:= ALLOCATED      !
         !      actual process:=newproc        !
                              !
                              !
            ! RESTORE CONTEXT (newproc) !
                              !
                              !
          ! UNLOCK (lockvariable of pool) !
                              !
                              !
                           (     )


                          ( RESIGN  )
                              !
                              !
          ! SAVE CONTEXT (actual process) !
                              !
                              !
       !     LOCK (lockvariable of pool)        !
       ! (*pool of the executing processor*)    !
                              !
                              !
   !        INSERT (actual process,readylist)          !
   !         state of actual process:=READY            !
   ! state of executing processor:=DEALLOCATED         !
   !           actual process:=NIL                     !
                              !
                              !
          ! UNLOCK (lockvariable of pool) !
                              !
                              !
                           (     )
```

```
                  ( BLOCK )
                      !
                      !
 !         SAVE CONTEXT (actual process)           !
 ! (*actual process of executing processor*)      !
                      !
                      !
 !        state of actual process:=BLOCKED          !
 ! state of executing processor:=DEALLOCATED       !
 !              actual process:=NIL                 !
                      !
                      !
                  (       )


                  ( RESUME )
                      !
                      !
      ! executing processor := ALLOCATED !
                      !
                      !
                  (       )


                  ( READY )             parameter: process
                      !
                      !
 !             LOCK (lockvariable of pool)                !
 !   (*pool of processors identified in the des-          !
 !   criptor of 'process', not necessarily ident-         !
 ! ical with the pool of the executing processor*)        !
                      !
                      !
         !  INSERT (process,readylist) !
         !   (*readylist of that pool*) !
                      !
                      !
           ! state of process:=READY !
                      !
                      !
        ! UNLOCK (lockvariable of pool) !
                      !
                      !
                  (       )
```

```
                    ( NOTIFY )      parameter: pool
                        !
                        !
 no   .-< DEALLOCATED processor in pool available >-.  yes
      !                                             !
      !                                             !
! select a processor, whose !       ! select a DEALLOCATED !
! 'actual process' has      !       ! processor            !
! lowest priority           !       !                      !
      !                                             !
      ----------------------------------------------
                        !
                        !
   ! send stimulus (*interrupt*) to this processor !
   !       state of this processor:=NOTIFIED       !
                        !
                        !
                   (          )
```

A.7. Preemption and Processor Redispatching

```
          ----
             /
          ---->( REDISPATCH  )
                     !
                     !                    no
    < state of executing processor=NOTIFIED >-->-
                     !                           !
                     !yes                        !
        no           !             yes           !
        -<--< switch processes  >-->-            !
        !                             !          !
        !                             !          !
   ! RESUME  !                   ! RESIGN  !     !
        !                             !          !
        !                             !-----<---
        !                             !
        !                        ! ASSIGN  !
        !                             !
        ------->-----------<-------
                     !
                     !
                  (      )----
                              /
                              ---->
```

```
PROCEDURE REDISPATCH;
BEGIN
IF processor.state=notified
THEN IF preempt (*conditions to redispatch processor are true*)
     THEN BEGIN resign;
                assign
          END
     ELSE       resume
ELSE assign
end
```

simplified version (c.f. par. 7, note 1):

```
               ----
                  /
                  ---->( REDISPATCH  )
                             !
                             !                         no
       < state of executing processor=NOTIFIED >-->-
                             !                         !
                             !yes                      !
                             !                         !
                        ! RESIGN   !                   !
                             !                         !
                             !-------------------------
                             !
                        ! ASSIGN   !
                             !
                             !
                         (       )----
                                     /
                                     ---->
```

```
PROCEDURE REDISPATCH;
BEGIN
IF processor.state=notified THEN resign;
assign
end
```

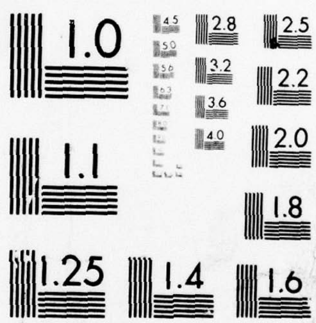
AD-A068 266 PURDUE UNIV LAFAYETTE IND PURDUE LAB FOR APPLIED IND--ETC F/G 9/2
MINUTES 1978 SPRING REGIONAL MEETING INTERNATIONAL PURDUE WORKS--ETC(U)
JUN 78 N00014-78-C-0127

UNCLASSIFIED NL

4 OF 4

AD
A068266

END
DATE
FILMED
6 -- 79
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

A. 9. SYNCHRONIZATION PRIMITIVES

Definition of data types used by the synchronization functions:

```
TYPE synchelement = RECORD
                    counter : integer;
                    pil     : list (* OF listelement *)
                    END;

TYPE listelement  = RECORD
                    proc    : processid;
                    info    : buffer
                    END;

TYPE list         = definition of an arbitrary list structure;

TYPE buffer       = definition of an arbitrary buffer or a
                    pointer to an arbitrary buffer;

TYPE processid    = RECORD
                    pool    : processorid;
                    (* other information *)
                    END;
TYPE processorid  = reference to processor pool descriptor;

VAR actproc : processid; (* identification of the running pro-
                           cess exists for each processor *)
```

Definition of elementary synchronization functions INC and DEC:

```
PROCEDURE INC (VAR syname : synchelement;
               VAR info   : buffer);
   VAR listel : listelement;
   BEGIN syname.counter := syname.counter + 1
      IF syname.counter <= 0 (* test for waiting process *)
      THEN BEGIN
           remove (listel,syname.pil); (* remove waiting process *)
           listel.info∧ := info; (* pass address of info to address
                                   specified by removed process *)
           ready (listel.proc);  (* ready the waiting process *)
           notify (listel.proc.pool); (* notify the corresponding
                                   processor-pool *)
           END
      ELSE insert ((actproc,info), syname.pil) (* no process
                                   waiting, insert the process
                                   executing the INC plus the
                                   address of the send info *)
END;

PROCEDURE DEC (VAR syname : synchelement,
               VAR info   : buffer);
VAR listel : listelement
BEGIN syname.counter := syname.counter - 1 (* decrement counter *)
  IF   syname.counter < 0   (* check if information available *)
```

```
  THEN BEGIN
       insert ((actproc,∧info),syname.pil); (* no info avai-
                lable, insert the process executing DEC plus the
                address of the info to be sent *)
       block;  (* block executing process *)
       assign; (* assign new process to processor *)
       END
  ELSE BEGIN
      remove (listel,syname.pil); (* info is available in the list *)
       info := listel.info∧; (* pass address of info to result
                                    parameter *)
       END
END
```

APPENDIX B OF TC 8 REPORT

PAPER LIST

CATEGORIES

I.....Up to Date Report

II....Working Papers

III...TC 8 Reference Papers (obtainable from TC 8)

IV....General Reference Papers

NUMBERING OF THE PAPERS

- category
- number within the category
- version of the paper (optional). *Higher version number automatically* replaces previous papers.

PAPER LIST

I-1-5..........TC 8
REPORT
October 1977

II-1-1.........Juergen Nehmer
On a Precise Operating System Terminology
July 1975

II-2-1.........Juergen Nehmer
Comparison of three Synchronizing Concepts
December 1975

II-3-1.........Gerhard Schrott
Common Elementary Synchronization Functions for an Operating System Kernel
January 1976

II-4-1.........Thierry Lalive d'Epinay
Notes to the Multiprocessor Concept
March 1976

II-5-1.........Tony Mark
On Completion of the Kernel
September 1976

II-6-1.........Drago Vojnovic
Monitor Synchronization Concept for I/O Processing

II-7-1.........Tony Mark
Exception Handling
January 1977

II-8-1.........Thierry Lalive d'Epinay
Implementation of an Operating System Kernel
June 1977

II-9-1.........Ivan Bach
On the Scheduling Problem in Real Time Operating Systems
September 1977

II-10-1........Jiri Hoppe and Thierry Lalive d'Epinay
Synchronization between Processes Running on Processors Without Common Memory

II-11-2........Gerhard Schrott
Additional Functions to the Operating System Kernel

III-1..........Thierry Lalive d'Epinay
Software-Organization for Process-Computers
June 1974

III-2..........Juergen Nehmer
Dispatcher Primitives for the Construction of Operating System Kernels
Acta Informatica 5, 237-255 (1975)

III-3..........H.G. Mendelbaum
For a Structuration Methology in Control Software Design
July 1974

III-4..........W. Kaiser
Overview on the Operating Systems Telemequanique RTDMS - DEC RT 11

JANUARY 1975

III-5..........VDI/VDE - GMR 4.2
Process Control Computer Operating Systems
June 1975

III-6..........Juergen Nehmer
Synchronizing Concepts and their Implementation by a Hierarchy of Elementary Operating System Functions

III-7..........Thierry Lalive d'Epinay
A New Method of Constructing and Using Real-Time Operating Systems
September 1975

III-8..........Gerhard Schrott
Definition and Construction of Real-Time Operating Systems by Basic Functions

III-9..........Ken Jackson et al
Mascot - A Modular Approach to Software Construction and Test
September 1975

III-10.........Ken Jackson
Modularity in Real-Time Computer Systems
August 1976

III-11.........Thierry Lalive d'Epinay
The Virtual Computer System
September 1976

III-12.........Ken Jackson
Language Design for Modular Software Construction
December 1976

III-13.........Tony Mark and Juergen Nehmer
High Level Interrupt Processing via Semaphores
September 1976

III-14-2.......Jiri Hoppe
Some Remarks Concerning I/O
March 1977

III-15.........TC 8
Comments on LTPL-Tasking Paper
March 1977

III-16.........T. Mark, O. Eggenberger, J. Nehmer
Experiences in the Implementation of a Structured Real-Time Operating System
1977

III-17.........J.J. Simon
Parallel Processing and High Order Languages
August 1977

III-18.........Drago Vojnovic
Kernel Real Time Systems
August 1977

IV-1...........Juergen Nehmer
Dispatcher Elementarfunktionen fuer symmetrische Mehrprozessor - DV-Systeme
September 1973

IV-2...........Juergen Nehmer
Ein Ansatz zur Standardisierung von Betriebssoftware
1973

IV-3...........N.G. Gammage
Virtual Processors for Real-Time Applications
April 1974

IV-4...........K.H. Timmesfeld
Joint Tasking Proposals for a LTPL
December 1975

PRECEDING PAGE BLANK - NOT FILMED